The Optimal Obstacle Placement With Disambiguation Problem

by

Polat Charyyev

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Mathematics

2017

The Optimal Obstacle Placement With Disambiguation Problem

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

Polat Charyyev

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Assoc. Prof. Atilla Yılmaz

_____

Assoc. Prof. Mine Çağlar

_____

Assoc. Prof. Elvan Ceyhan

_____

Asst. Prof. Özgür Asar

_____

Asst. Prof. Ümit Işlak

Date: _____

This thesis is dedicated to my parents.

ABSTRACT

In the optimal obstacle placement with disambiguation (OPD) problem, we investigate how traversal length depends on the spatial pattern of the obstacles. In the OPD problem an obstacle placing agent (OPA) wishes to insert (disk-shaped) obstacles of two types as true or false obstacles in an environment so as to maximize the traversal length of a navigating agent (NAVA). NAVA is equipped with a sensor that can only assign probabilities to each obstacle as being a true obstacle, but does not know the actual status of the obstacle until it reaches the boundary of the obstacle. When NAVA comes by the obstacle, it disambiguates the status of the obstacle with a cost added to the traversal length. We first investigate the case when OPA equips the overall working window with false and/or true obstacles together where the whole obstacle pattern changing from uniformness to regularity and from uniformness to clustering, then on the average, the mean traversal length tends to increase (decrease) as the obstacle pattern changes from uniformness to regularity (clustering). Secondly, we introduce M2^k algorithm which is the modified version of RD algorithm and mainly based on the effective choice of a subset of possible disambiguations. We observed that the trends for mean traversal length estimated by M2^k algorithm is similar to RD algorithm except for reducing the complexity time and keeping relative error less than 2.5%. Moreover, we study the case when obstacles are distributed inside various obstacle window types such as linear strips, V-shaped, semicircular, and elliptical where we change the parameters such as location, orientation and curvature of these obstacle window forms. We also consider the case where the true obstacles are placed randomly proportional to the areas of Voronoi polygons or Delaunay triangles based on the allocation of the clutter points. On the average, the elliptical window type (a continuous version of V-shaped) reveals to be the best

performing in maximizing the traversal length of NAVA and as for tessellations, the mean traversal length is essentially the same as the mean traversal length when obstacles are distributed uniformly. On the other hand, we provide extensions of the existing sub-optimal algorithms in the OPD problem, and introduce new ones for NAVA. The goal is to find an algorithm to minimize the expected traversal length of NAVA. We generalize the penalty-based functions used within the existing heuristic algorithms, and combine possibly related algorithms in a single family. We provide a guidance for choosing the algorithm from the defined family when the performance of NAVA's sensor varies from poor to almost perfect detection (of true obstacles). Our results are supported by extensive Monte Carlo simulations and theoretical results for some special types of graph structures are also provided.

ÖZETÇE

Belirsizliği giderme özellikli optimal engel yerleştirme (OPD) probleminde, yol uzunluğunun uzaysal desen türlerine göre nasıl etkilendiğini araştırıyoruz. OPD probleminde, navigasyon aracının (NAVA) gideceği yolu mümkün olduğunca uzun tutmak için yüzeye engel yerleştirme aracı (OPA) tarafından gerçek veya sahte engeller yerleştirilmektedir. NAVA her diskin gerçek engel olma ihtimalini belirleyen sensör ile donatılmış olup, disklerin gerçek veya sahte olduğunu diskin sınırlarına gelene kadar bilmemektedir. Fakat NAVA diskin sınırlarında iken belirli bir maliyetin toplam süreye/uzunluğa eklenmesi karşılığında disklerin gerçek veya sahte olduğunu öğrenebilmektedir. ?lk olarak, OPA'nın çalışma bölgesine sahte ve/veya gerçek engelleri tüm engel desenin tekdüzelikten düzenliliğe ve tekdüzelikten kümelenmeye göre yerleştirilmesi sonucunda toplam geçiş uzunluğunun engel desenin tekdüzelikten düzenliliğe göre artmakta ve tekdüzelikten kümelenmeye göre ise azalmaktadır. İkinci olarak, RD algoritmasının değiştirilmiş versiyonu olan ve temel olarak disk başına düşen belirsizlik nokta sayısının akıllıca seçilmesine bağlı olan $M2^k$ algoritmasını tanıtmaktayız. $M2^k$ algoritması tarafından hesaplanan yol uzunluğunun RD algoritması tarafından hesaplanan yol uzunluğu ile benzer eğilimler gösterdiğini gözlemledik ve dahası hesaplama süresi azaltılmış olup, toplam geçiş uzunluğu en fazla %2.5 sapmaktadır. Buna ek olarak, engellerin lineer, V-şeklinde, yarı çembersel, ve eliptik gibi değişik şekil içinde dağılımları engel şekillerin lokasyon, oryantasyon, ve kurvatür gibi parametrelerine göre incelenmiştir. Ayrıca, sahte engellerin yerleşimine bağlı olarak Voronoi çokgen veya Delaunay üçgen alanlarına orantılı olacak şekilde gerçek engellerin yerleştirilme durumu ele alınmıştır. Ortalama olarak, NAVA'nın toplam geçiş uzunluğunu maksimuma çıkarmak için engel şekillerin arasından eliptik engel şekli (V-şeklinin sürekli versiyonu) en optimal değeri vermektedir ve Dirichlet

mozaiği gibi düzenlemeler için ise geçiş uzunluğu engellerin tekdüze dağılımındakine yakın değeri vermektedir. Diğer taraftan, OPD problemindeki NAVA için mevcut optimal olmayan algoritmaları genişletmekteyiz ve yenilerini ortaya koymaktayız. Buradaki asıl amaç, NAVA'nın beklenen geçiş uzunluğunu minimuma indiren algoritma geliştirmektir. Mevcut sezgisel algoritmalarda kullanılan ağırlığa dayalı fonksiyonları genelleştiriyoruz, ve benzer algoritmaları tek çatı altında birleştiriyoruz. NAVA'nın sensör algılama (gerçek engel olma ihtimali) performans gücünün en zayıftan mükemmel dereceye kadar durumlarda birleştirilen algoritma kümesinden en optimal olanı seçmeyi gösteriyoruz. Bizim sonuçlarımız kapsamlı Monte Carlo simülasyonları ile desteklenmekle beraber, bazı özel graf biçimleri için teorik sonuçlar göstermekteyiz.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisors Assoc. Prof. Elvan Ceyhan and Assoc. Prof. Atilla Yılmaz for the continuous support of my Ph.D study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank Assoc. Prof. Vural Aksakallı for his guidance and help on designing and understanding the related algorithms. I thank my thesis committee members Assoc. Prof. Atilla Yılmaz and Assoc. Prof. Mine Çağlar for their insightful comments and encouragement. I also thank thesis jury members Asst. Prof. Özgür Asar and Asst. Prof. Ümit Işlak for accepting and participating my thesis defense.

I thank my Ph.d colleagues, especially Artür Manukyan for the stimulating discussions and for all the fun we have had in the last five years. I thank Mathematics department Professors for creating academic environment and all GSSE staff, especially Emine Büyükdurmuş and Elif Tüysüz for their continual support and guidance. I would like to thank Koç University and TUBITAK for providing me with financial support which made this study possible.

I would like to thank my family: my parents, my brother and sisters, my wife and beloved son for supporting me spiritually throughout writing this thesis and my life in general.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Abbreviations

CTP Canadian Traveler's Problem

D-SOS Discretized Stochastic Obstacle Scene

DTA Distance to Termination Algorithm

NAVA Navigating Agent

OPD Obstacle Placement with Disambiguation

OPA Obstacle Placing Agent

RDA Reset Disambiguation Algorithm

SOS Stochastic Obstacle Scene

SRA Simulated Risk Algorithm

Chapter 1

INTRODUCTION

The optimal obstacle placement with disambiguation (OPD) problem was first introduced by Aksakalli and Ceyhan (2012). The OPD problem is the dual of the stochastic obstacle scene (SOS) problem (Papadimitriou and Yannakakis, 1991), where the discrete version is called the Canadian traveler's problem (CTP) (Bar-Noy and Schieber, 1991). In the SOS problem, the main goal is finding the shortest path through an unknown environment wherein a navigating agent (NAVA) plans the route from a given starting point to a designated target point through randomly positioned disk shaped regions that represent obstacles. In the version of the OPD problem of Aksakalli and Ceyhan (2012), the navigation environment already contained clutter, i.e., false natural obstacles, and an obstacle placing agent (OPA) placed true obstacles in the same environment. In practice this setting may arise in various situations, for example, OPA can represent a naval warfare vehicle placing mines (i.e., true obstacles) on the surface of a water body (such as a sea or ocean) which already has natural or artificial obstacles (like rocks or debris). And, NAVA could be a battleship (from the opponent forces to OPA) and wants to traverse from a source point to a target point which might be located at the coast defended by OPA and its affiliates. Prior to NAVA's traversal, NAVA obtains the respective probabilities of the obstacle disks being true obstacles by using its sensor. Throughout the traversal, when NAVA reaches a disk's boundary, it can disambiguate the disk and learn the actual status of it, i.e., whether it is a true or false obstacle. The cost of the disambiguation for each disk is added to the total traversal length (which contributes to the total cost in our optimization problem). If the disambiguated disk is a false obstacle, then NAVA can pass through

or over it; otherwise NAVA has to replan its route from the current location avoiding the true obstacles until it reaches the target. From NAVA's perspective, the task is then finding the optimal path for reaching from the source point to the target, i.e., the path with *minimum* expected traversal length. Conversely, the OPA's goal is placing the obstacles in a such a way that NAVA's traversal length is *maximized*. Here, we assume that NAVA uses a greedy algorithm called the RD algorithm (Aksakalli et al., 2011) in choosing the optimal path. Hence, NAVA disambiguates any obstacle it runs into and then always seeks a better policy and updates the algorithm accordingly. On the other hand, OPA will search for a better placement of obstacles. We also consider the version of the OPD problem where OPA has the capability of inserting both true and false obstacles in the navigation environment. Although these optimization problems have gained considerable attention in literature, few theoretical and efficient optimal results have been found. In particular, Papadimitriou and Yannakakis (1991) showed that the SOS problem is NP-hard, and as for the CTP with its variants, one can find many heuristics for special cases wherein the optimal solution is attained in some special graph structures (Nikolova and Karger, 2008; Xu et al., 2009; Bnaya et al., 2009; Fried, 2013). Among these references, the work of Bnaya et al. (2009) is the closest to ours. They introduce the sensing cost of an edge in the CTP problem which is equivalent to the disambiguation cost of an obstacle in the SOS problem. In this variant of CTP, additional actions, called *remote sensing*, are allowed where each such action reveals the status of a non-incident edge for a certain cost. Indeed, allowing remote-sensing makes CTP NP-hard even on disjoint-path graphs (Fried et al., 2013).

Regarding the suboptimal algorithms for SOS problem, the BAO* algorithm of Aksakalli (2007a), the simulated risk disambiguation (SRA) of Fishkind et al. (2007) and the reset disambiguation algorithm (RDA) of Aksakalli et al. (2011) are among the examples. Indeed, the RD algorithm is an efficient algorithm for discretized SOS problem and yields an optimal value for graphs with two nodes (called *parallel* graphs $G = (V, E)$ with $|V| = 2$), and for disjoint-path graphs. In general, the discrete SOS

problem for *parallel* graphs can be solved in $O(|E|!)$ time complexity, but using the policy dictated by the RD algorithm the complexity of solving the same problem reduces to $O(|E| \log |E|)$. Successful adaptations of all these heuristics to the discretized SOS problem and the generalizations of both SRA and RDA were given in Aksakalli and Ari (2014), which in turn is called the "distance to termination" (DT) algorithm. The DT algorithm exploits the distance from current position of any obstacle to the target point together with the corresponding probabilities of obstacles being true obstacles.

As for CTP, several variants of it have been introduced and effective heuristic algorithms are proposed. For example, in Bar-Noy and Schieber (1991) the *recoverable* CTP was introduced where a blocked edge does not remain blocked forever, and each vertex v has a *recovery* time $\ell(v) \in [0, \infty)$ for edges adjacent to v. They also introduced the k-CTP, in which an upper bound of k blocked edges is given. Nikolova and Karger (2008) studied another variant of the CTP where the cost of edges comes from independent and identical distributions. Bnaya et al. (2008) introduced the sensing cost in CTP which is closely related to our work. In the original CTP, the status of an edge (i.e., whether it is blocked or not) can only be revealed upon reaching a vertex incident to that edge. This kind of sensing is called the *local* sensing, and the *remote* sensing refers to revealing the status of an edge from a distant location (Bnaya et al., 2009). The sensing cost in the CTP is equivalent to the disambiguation cost in SOS problem. Bnaya et al. (2015) studied the repeated-task CTP where a number of navigating agents need to travel with the same goal, minimizing their combined travel cost. Moreover, recently Aksakalli et al. (2016) developed an AO* based exact algorithm, called the CAO* algorithm, for the CTP and they showed that the empirical performance is better than other exact algorithms.

1.1 Preliminaries

The most common notion of comparison of random variables is the stochastic order. For any real valued random variable X, let $F_X(x) = \mathbf{P}(X \leq x)$ be the cumulative distribution function of X.

Definition 1.1.1 *If X and Y are random variables defined on the same sample space Ω, then X is stochastically smaller than Y ($X \leq_{st} Y$) if $F_X(\omega) \leq F_Y(\omega)$ for each $\omega \in \Omega$.*

Similar definition holds for '$\geq_{st}$'. There are many equivalent definitions describing the stochastic ordering of random variables. The following is the most common way of characterizing the stochastic ordering.

Definition 1.1.2 *If X and Y are random variables defined on the same sample space Ω, then $X \leq_{st} Y \iff \mathbf{E}f(X) \leq \mathbf{E}f(Y)$ for all non-decreasing functions f.*

Where $\mathbf{E}f(X)$ represents the expected value of $f(X)$ with respect to the density function of random variable X. Analogous definition holds for '$\geq_{st}$'.

Definition 1.1.3 *Let X_n be a sequence of random variables defined on a sample space Ω. We say that X_n is convergent in probability to a random variable X defined on Ω if and only if*

$$\lim_{n\to\infty} \mathbf{P}\left(|X_n - X| > \epsilon\right) = 0$$

for any $\epsilon > 0$. X is called the probability limit of the sequence and the convergence in probability is denoted by

$$X_n \xrightarrow{P} X.$$

1.2 Continuous SOS and OPD Problems

The continuous SOS problem is formally defined as follows: Consider a bounded obstacle field $\Omega \subset \mathbb{R}^2$. False obstacles (such as rocks, debris, fake mines) are assumed to be located at points, X_C, from a spatial point process C. Then, an Obstacle Placing Agent (OPA) places true obstacles at the points, X_O, where centers are from another spatial point process O, possibly different from C. All false obstacles (respectively true obstacles) are assumed to be disk shaped and centered at the points in X_C (respectively X_O). A Navigating Agent (NAVA) wishes to traverse from a given starting point, $s \in \Omega$, to a target point, $t \in \Omega$, and is equipped with a sensor that assigns random probabilities $p_C : X_C \to [0,1]$ and $p_O : X_O \to [0,1]$ prior to NAVA's traversal. When observing a realization of these processes, the NAVA only sees $X := X_C \cup X_O$ as obstacles, some of which maybe true obstacles. Let $p(x)$ be the probability that $x \in X_O$, that is, x is a true obstacle for all $x \in X$. Then NAVA's detector assigns $p(x) = p_O$ for $x \in X_O$ and $p(x) = p_C$ for $x \in X_C$. For every disk center x, the obstacle disk D_x is an open region with a fixed radius $r > 0$. The NAVA seeks to traverse a continuous (s,t) curve without hitting any true obstacle at shortest achievable arc length (i.e., in shortest traversal length). We further suppose that there is a dynamic disambiguation capability. Specifically, for all $x \in X$, when the traversal curve is on the boundary of the disk, denoted as ∂D_x, (i.e., NAVA comes right at the boundary of the obstacle), it has an option to disambiguate x, that is, to learn $x \in X_O$ or not, but at a cost $c > 0$ added to the length of the traversal curve. The NAVA can pass through disks that have been disambiguated and found to be clutter (i.e., false obstacle), but has to avoid ambiguous disks as well as disks that have been disambiguated and found to be true obstacles. How the NAVA should route the continuous (s,t) traversal curve - and where and when the disambiguations should be performed - in order to minimize the expected length (including the disambiguation cost) of this curve is called the *continuous SOS problem* (Papadimitriou and Yannakakis, 1991).

On the other hand, the problem of placing the obstacles so as to maximize the

NAVA's traversal length in the SOS problem, namely the optimal *obstacle placement with disambiguations* (OPD) problem was introduced by Aksakalli and Ceyhan (2012). Indeed, the OPD problem can be formulated as

$$\max_{X_O} \mathbf{E}\left[L(X_O \cup X_C)\right] \text{ such that } X_O \cap W^c = \emptyset,\ |X_O| = n,$$

where $L(X)$ is the traversal length of NAVA given a set of hindrance (obstacle or clutter) points of X and $W \subset \Omega$ is the window where obstacles are placed by the OPA.

1.3 Discretized SOS and OPD Problems

For computational efficiency and convenience, we consider a discrete approximation of the continuous setting on the 8-adjacency integer lattice as in Aksakalli et al. (2011). Specifically, this discretization is stored as a graph G whose vertices are all of the pairs of integers i, j such that $1 \leq i \leq i_{max}$ and $1 \leq j \leq j_{max}$ where i_{max} and j_{max} are given integers. The obstacle field Ω is partitioned by an $i_{max} \times j_{max}$ grid consisting of unit squares and the vertices of the squares in the grid constitute the vertices of G. The edges of G are determined by the adjacency of the vertices in the grid. That is, there are edges between all pairs of the following four types of vertices: (1) (i, j) and $(i+1, j)$ with unit length, (2) (i, j) and $(i, j+1)$ with unit length, (3) (i, j) and $(i+1, j+1)$ with length $\sqrt{2}$, and (4) $(i+1, j)$ and $(i, j+1)$ with length $\sqrt{2}$ for $i, j = 0, 1, \ldots, 99$. Also, we add corner edges of unit length with pairs of vertices connecting $(100, j)$ and $(100, j+1)$, and vertices connecting $(i, 100)$ and $(i+1, 100)$ for $i, j = 0, 1, \ldots, 99$. This discretization is called the 8-adjacency integer discretization of Ω. One vertex in G is designated as the starting point, s, and another vertex in G is designated as the target point, t (starting and target points are totally arbitrary). The NAVA wishes to traverse from s to t in G, using only the edges that do not intersect any true obstacle or ambiguous disks in the environment. If an edge intersects any ambiguous disk, then a disambiguation of the disk may be performed from either endpoint of the edge that are outside of the disk. The goal is to devise an algorithm (to be implemented

by NAVA) that minimizes its expected traversal length by effecient exploitation of the disambiguation capability and the information on the locations of the disks. We call this discretization as the *discretized SOS problem* (D-SOSP), which in effect, is a special case of the *Canadian traveler's problem*.

1.4 Organization of the Thesis

The rest of this thesis is organized as follows: In Chapter 2 we discuss choosing the optimal obstacle pattern against a disambiguating navigating agent. In Chapter 3 we introduce M2^k algorithm for the OPD problem and investigate the trends for mean traversal length by statistical analysis tools such as repeated measures ANOVA. In Chapter 4 we investigate how the traversal length depends on obstacle window types and tessellations. In Chapter 5 we consider heuristic algorithms for the OPD problem and provide a guidance for choosing the best performing algorithm given the problem setting. In Chapter 6 we present some theoretical results in the discrete SOS problem for some special graphs and present directions for future research. In Chapter 7 presents conclusions and discussion.

Chapter 2

CHOOSING THE OPTIMAL OBSTACLE PATTERN AGAINST A DISAMBIGUATING NAVIGATION AGENT

2.1 Introduction

In Aksakalli and Ceyhan (2012), NAVA uses the RD algorithm (Aksakalli et al., 2011) to find the shortest path and authors mainly focus on two problems. First, given a background clutter pattern, what would be the optimal number of obstacles and the obstacle pattern to use so as to maximize the NAVA's total traversal length. Second, if OPA is given a certain number of obstacles, then what would be the optimal way to place these obstacles among the given clutter locations. OPA places the true obstacles uniformly inside various obstacle windows such as linear strips, V-shaped and W-shaped windows together with the possibly natural pattern scenarios of false obstacles with different pattern types such as uniform (homogeneous Poisson process), clustered (Thomas and Matérn) and regular (hardcore and Strauss point process) patterns (Baddeley et al., 2008). Aksakalli and Ceyhan (2012) provided same evidence that on the average the traversal length of NAVA increases as the level of regularity increases and decreases as the level of clustering increases. In overall comparison, the mean traversal length of NAVA attains its maximum value when the background clutter type is hardcore, and the obstacle form is V-shaped with 50 uniformly sampled obstacles.

In this section, we investigate how traversal length depends on obstacle pattern changing from uniformness to regularity and from uniformness to clustering in a more detailed fashion. For regularity, the effects of two parameters, the pairwise distance between disk shaped obstacles and the number of such obstacles (i.e., its

density) are investigated. As for clustering, the effects of the radius in which obstacles accumulate around one or more cluster centers and the number of such cluster centers are investigated. In Aksakalli and Ceyhan (2012) a few clutter types were considered, but here we address more types of obstacle patterns and also study the case where OPA can insert both false and true obstacles.

2.2 Experimental Setting

For simplicity, we take the study window (i.e., the navigation environment) to be $\Omega = [0, 100] \times [0, 100]$ and we insert a grid of unit squares on the study window, so that we partition the window with 10^4 such squares. In each square we also insert the diagonal line segments connecting nonadjacent vertices. We represent this square grid partition (together with diagonals) as a graph $G = (V, E)$ such that V is comprised of vertices of the squares in the grid and edges are the edges of the squares together with diagonals. Hence, we have edges between all pairs of the following four types of vertices : (1) (i, j) and $(i + 1, j)$ with unit length, (2) (i, j) and $(i, j + 1)$ with unit length, (3) (i, j) and $(i + 1, j + 1)$ with length $\sqrt{2}$, and (4) $(i + 1, j)$ and $(i, j + 1)$ with length $\sqrt{2}$ for $i, j = 0, 1, \ldots, 99$. Also, we add corner edges of unit length with pairs of vertices connecting $(100, j)$ and $(100, j + 1)$, and vertices connecting $(i, 100)$ and $(i + 1, 100)$ for $i, j = 0, 1, \ldots, 99$. This discretization is called the 8-adjacency integer discretization of Ω. Without loss of generality, we take a starting point s on the upper boundary of the study window, say $s = (50, 100)$. And, for the target we take a point on the opposite side of rectangular window, say $t = (50, 1)$. In practice, we can think of Ω as a section of sea near the coast which is going to be defended by OPA, where NAVA could be a ship trying to navigate over the sea to reach the target at the coast.

Here, OPA inserts disk shaped obstacles whose centers are generated from a random point process. In this version, OPA determines only the distribution of the obstacles, but not their exact locations. Each disk is either traversable (i.e., false or clutter obstacle) or non-traversable (i.e., true obstacle) with their respective proba-

bilities of being a true obstacle. In the OPD problem an obstacle is traversable if it is of clutter type and non-traversable if it is a true obstacle. The traversable (respectively, non-traversable) obstacles, i.e., clutter (respectively, true) obstacles, are denoted with a set of circles whose centers are the set of points denoted by X_C (respectively, X_O). NAVA wishes to traverse from the given starting point s to the target point t inserted with a sensor assigning (random) probabilities $p_c : X_C \to [0, 1]$ and $p_o : X_O \to [0, 1]$ prior to its traversal. That is, the sensor assigns the status of disks with the probability function $f(x)$,

$$f(x) = \begin{cases} p_c, & x \in X_C \\ p_o, & x \in X_O \\ 0, & \text{otherwise} \end{cases}$$

where p_c and p_o are also random variables with density functions F_C and F_O, respectively. The sensor would (preferably) attach high probabilities for true obstacles compared to clutter. NAVA knows the exact locations of $X = X_C \cup X_O$, but not the actual status of disks, i.e., whether a disk is traversable or not, but only have a probability which is suggestive of the actual status of the disks. Also suppose that when the NAVA is located on the boundary of a disk, it disambiguates a disk at a cost $c > 0$ added to the traversal length; i.e., NAVA can learn the actual status of the disk with the specific cost c. NAVA disambiguates the obstacle and if the obstacle is of clutter type, NAVA is able to traverse through it and can not traverse otherwise. Each disk has a fixed radius of $r = 4.5$ units and the disk centers are sampled inside $[10, 90] \times [10, 90]$ ensuring that there always exists a (possibly very long) s, t walk. The existence of such a long traversable path is based on the assumption that NAVA must reach the target, but in an untimely manner (i.e., traversal length being larger than a threshold) and for larger traversal times/lengths NAVA's arrival at the target (is assumed to) pose no threat to defending entities (including OPA). That is, NAVA's arrival on time to the target could make a difference for the state of affairs and if arrival is sufficiently late, then we assume it is irrelevant from OPA's perspective. The disambiguation cost of a disk is taken to be $c = 5$, and in each part clutter proba-

bilities p_c are from Beta$(2, 6)$ distribution, whereas true obstacle probabilities p_o are from Beta$(6, 2)$ distribution. From NAVA's perspective, small probabilities indicate that it is more likely to be traversable, whereas high probabilities show that it is less likely to be traversable. The means of random variables sampled from Beta$(2, 6)$ and Beta$(6, 2)$ are 0.25 and 0.75, respectively, which resonates with the idea that the sensor assigning high probabilities to true obstacles. So, it is reasonable to work with these skewed distributions for sensor's capability of obstacle detection. These choices (of the cost c and sensor probability distributions) are made in line with those of Aksakalli and Ceyhan (2012) and Priebe et al. (2005). However, if one wants to increase the sensor's sensitivity, one might pick Beta(α, β) with $\alpha < 2, \beta > 6$ for false obstacles and Beta(α, β) with $\alpha > 6, \beta < 2$ for true obstacles.

2.3 Obstacle Pattern Ranging from Uniformness to Regularity

In the OPD problem, we investigate how the traversal length varies as the obstacle pattern changes. First, we equip the whole window with only false or only true obstacles whose pattern changes from uniformness to regularity. We denote the number of clutter (false obstacles) by n_c and the number of true obstacles by n_o. We compute the traversal lengths for $n_c = 25, 50, 75, 100$ and $n_o = 25, 50, 75, 100$. For inserting obstacles conforming to regularity, the points that constitute the centers of disk shaped obstacles are chosen from the Strauss process (Baddeley et al., 2008), denoted as Strauss(n, d, γ), where n stands for the number of points, γ is the intensity of the number of pairs of distinct points lying closer than d units; i.e., the parameter γ controls 'strength' of interaction between points. For any fixed value of d, if $\gamma = 1$ the density reduces to a uniform point process (i.e., the pattern becomes uniformness), and if $\gamma = 0$ the density is a hard core process (i.e., completely regular pattern). For values $0 < \gamma < 1$, the process exhibits negative association (i.e., inhibition) between points. Thus, in our simulations we choose the γ values from 0 to 1 with an increment size of 0.1 units and we also consider various values of d varying from 0.5 to 11 with an increment size of 0.5 units. Moreover, the whole window is inserted with both

false and true obstacles together where the whole pattern changes from uniformness to regularity for $n_c = 25, 50, 75$ and $n_o = 25, 50, 75$ such that $n_c + n_o = 50, 75, 100$.

In our data set γ, d and the number of obstacles (hence implicitly obstacle density) are the main factors that influence the traversal length. In Aksakalli and Ceyhan (2012), in the regularity pattern Strauss(n_c, d, γ), the number of clutter was fixed to be $n_c = 100$, d was taken to be 5 and γ values were 0, 0.5, and 1. And, the number of true obstacles n_o was taken to be 20, 30, 40, 50, and 60. They observed that the NAVA tends to have higher traversal lengths for smaller γ values. In the current study, we want to address how the traversal length changes when both d and γ change under regularity by considering various values of d and γ simultaneously. Based on our Monte Carlo simulations, on the average the traversal lengths tend to increase as the obstacle pattern changes from uniformness to regularity.

2.3.1 Clutter Only Case

In this section, we consider the case of only clutter type obstacles being inserted into the study window where obstacle pattern ranges from uniformness to regularity. Since

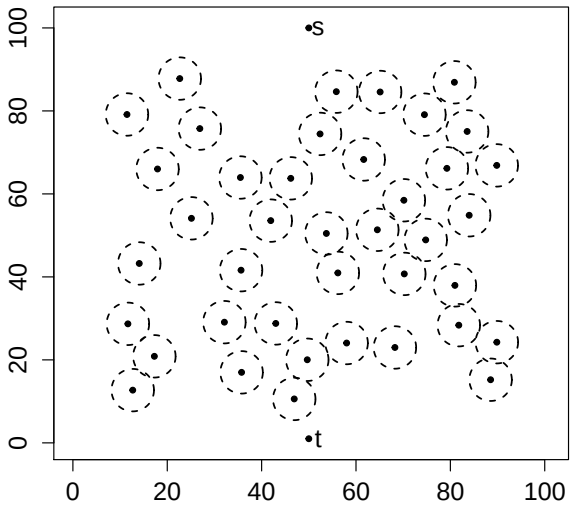


Figure 2.1: A realization of clutter disks (dashed circles) with $n_c = 40$ from the Strauss$(n_c = 40, d = 9, \gamma = 0)$ process.

there are 11 different values of γ, 22 different values of d and together with 100 Monte Carlo replication for each clutter number level n_c, we have $11 \times 22 \times 100 = 24200$ measurements. For each realization, the traversal length is computed by using RD algorithm described in Aksakalli et al. (2011). An example for the realization of this setting is given in Figure 2.1.

We investigate the trends in the mean traversal length of NAVA as a function of the level of regularity of clutter points. First, we only consider the influence of γ values (ignore d values) on the traversal length for each clutter number level. We model the traversal length in this scenario as follows,

$$Y_{ij} = \mu_j + \beta_j X_{ij} + \epsilon_{ij} \tag{2.1}$$

where $i = 1, 2, ..., 24200$ and $j = 1, 2, 3, 4$; Y_{ij} (i.e., the dependent variable) is the i^{th} traversal length of NAVA at j^{th} clutter level, μ_j is the mean for clutter number level j, β_j is the rate of change in mean traversal length in clutter number level j, X_{ij} (continuous co-variate) is the corresponding gamma value of i^{th} observation at j^{th} categorical group, and ϵ_{ij} is the associated error term. Since n_c takes a few values,

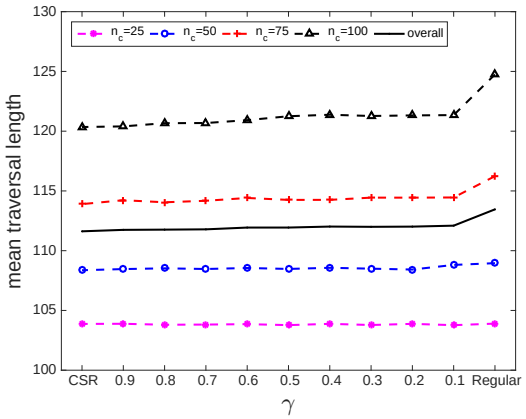


Figure 2.2: Mean traversal length versus γ values for groups $n_c = 25, 50, 75, 100$ together with overall plot (i.e., the average of all groups combined).

we take the number of clutter levels as a categorical variable rather than a numerical

variable. Recall that for any fixed d, when $\gamma = 1$ the point pattern is uniform and when $\gamma = 0$ the point pattern becomes completely regular. In Figure 2.2, we provide interaction plots for different clutter number levels versus γ values and also overall plot of the average of all groups versus γ values. In Figure 2.2, when the point pattern changes from uniformness to regularity the average traversal lengths seem to remain unchanged or increase slightly for all clutter number levels. We record the intercept and slope values of regression lines for each clutter number level n_c (see Table 2.1). So, using ANCOVA (Seltman, 2012) with model given in Equation (2.1), we find

	$n_c = 25$	$n_c = 50$	$n_c = 75$	$n_c = 100$
intercept	103.84	108.36	113.79	119.96
slope	0	0.03	0.12	0.26

Table 2.1: Intercept and slope values of regression lines for each categorical group n_c.

out that there is a significant interaction between clutter number levels and γ values (p-value$<$0.001). Hence, not all β_j's are equal, i.e., the lines in Figure 2.2 are not all parallel to each other although the lines in the plot suggest otherwise. The next

intercept

	$n_c = 50$	$n_c = 75$	$n_c = 100$
$n_c = 25$	$<$0.001	$<$0.001	$<$0.001
$n_c = 50$		$<$0.001	$<$0.001
$n_c = 75$			$<$0.001

(a)

slope

	$n_c = 50$	$n_c = 75$	$n_c = 100$
$n_c = 25$	0.003	$<$0.001	$<$0.001
$n_c = 50$		$<$0.001	$<$0.001
$n_c = 75$			$<$0.001

(b)

Table 2.2: Significant level of differences (p-values) in intercept (a) and slope (b) for each pair of groups.

natural question would be which clutter number levels have significantly different

trends than other(s). To explain this, we repeat the model in Equation (2.1) for each pair of groups to get the pairwise significance level of differences in intercept and slope (see Table 2.2). From Table 2.2 it follows that both intercepts and slopes for clutter number levels 25, 50, 75 and 100 are significantly different from each other. Thus, the lines in Figure 2.2 have significantly different slopes except for $n_c = 25, 50$.

So far, we have not taken into consideration the influence of the d values. Figure 2.3 shows the plots of the average traversal length versus d values. Observe that the

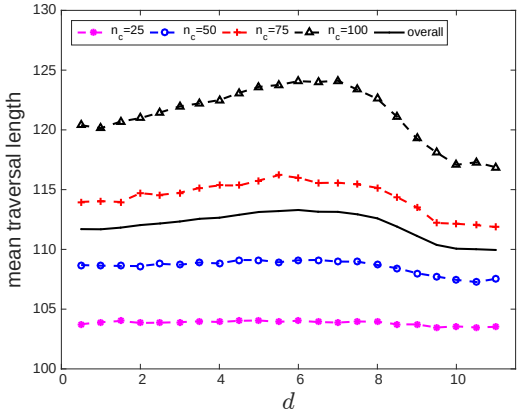


Figure 2.3: Mean traversal length versus d values for groups $n_c = 25, 50, 75, 100$ with overall plot (i.e., the average of all groups combined).

change in d values influences the traversal lengths considerably for larger n_c values. The Figure 2.3 also suggest choosing $n_c \geq 75$ and $d \approx 6$ for larger traversal length. We fit the data set into the best quadratic polynomial because the curves in Figure 2.3 suggest that there is a concave down trend at least for $n_c = 75$ and 100. In Figure 2.4, we provide graphs separately for each clutter number level and show corresponding fitted quadratic polynomial. Clearly, the concave down trend gets more emphasized as the number of clutter increases. Also, for any fixed value of γ and when d ranges from 0 to 8 the obstacle pattern gets more regular and hence yields higher traversal length. But, when d is larger than 9 units the average traversal length decreases sharply because the obstacle pattern leaves obstacle-free spaces in the region through

which NAVA can quickly traverse without encountering any obstacle. These specific

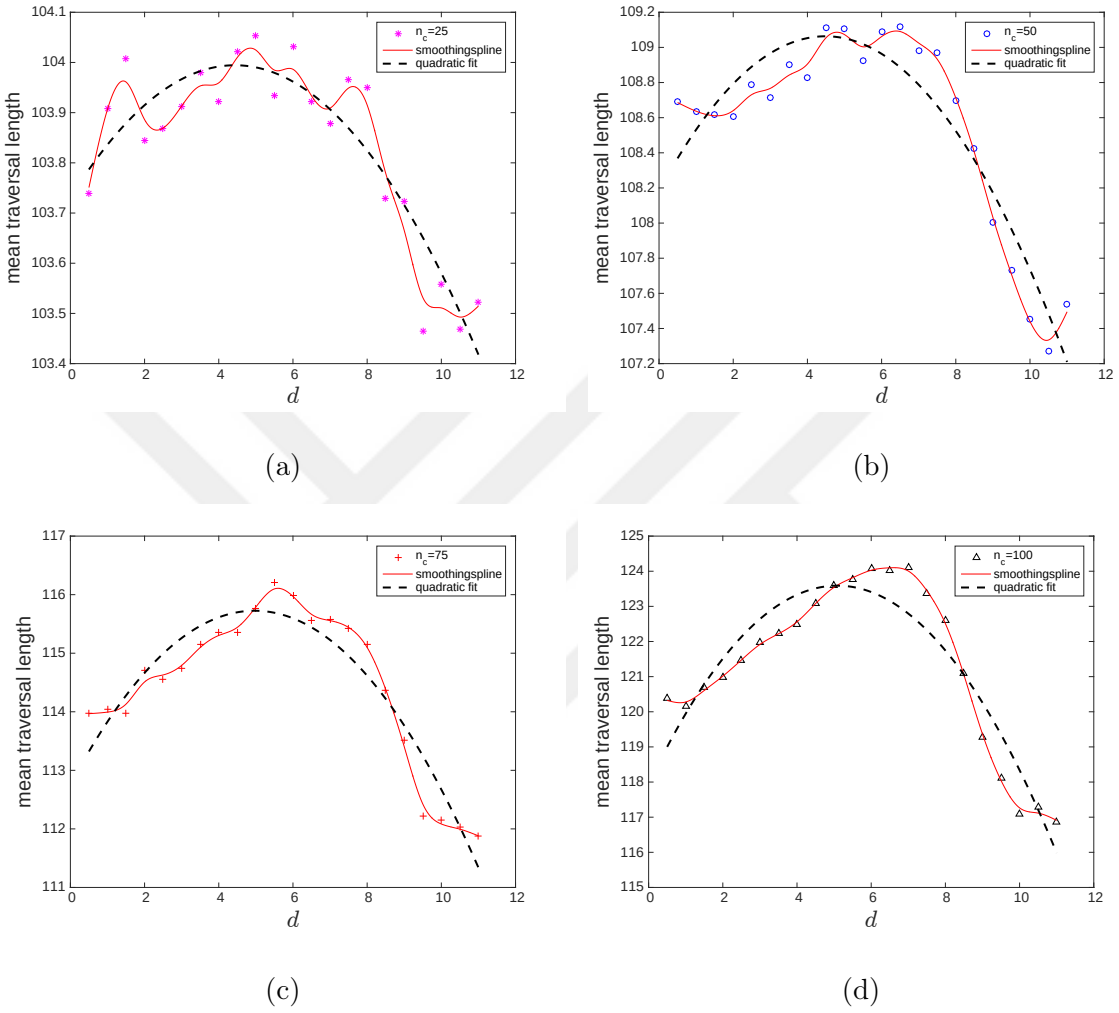


Figure 2.4: Mean traversal length versus d values for $n_c = 25, 50, 75, 100$ together with quadratic polynomial fit and (nonparametric) smoothing spline with smoothing parameter $p = 0.9863$. Notice that vertical axes are differently scaled.

threshold values of d are closely related to the radius of disk shaped obstacle (i.e., $r = 4.5$ units). When $d = 9$, recalling that $r = 4.5$, the disk-shaped obstacles become tangent to each other. Thus, for values of d larger than 9 the obstacles are no longer tangent and, so distances between obstacles increase resulting in a shorter path.

Regarding the plots in Figure 2.4, we consider the following model for the traversal

coefficients \ groups	$n_c = 25$	$n_c = 50$	$n_c = 75$	$n_c = 100$	overall
α	103.73	108.73	112.75	117.95	110.65
β	0.12	0.39	1.20	2.20	0.98
δ	-0.01	-0.04	-0.12	-0.21	-0.10

Table 2.3: Coefficients of quadratic polynomial fitted to the data at each clutter number level.

intercept

	$n_c = 50$	$n_c = 75$	$n_c = 100$
$n_c = 25$	<0.001	<0.001	<0.001
$n_c = 50$		<0.001	<0.001
$n_c = 75$			<0.001

(a)

linear term

	$n_c = 50$	$n_c = 75$	$n_c = 100$
$n_c = 25$	0.17	<0.001	<0.001
$n_c = 50$		<0.001	<0.001
$n_c = 75$			<0.001

(b)

quadratic term

	$n_c = 50$	$n_c = 75$	$n_c = 100$
$n_c = 25$	0.07	<0.001	<0.001
$n_c = 50$		<0.001	<0.001
$n_c = 75$			<0.001

(c)

Table 2.4: Significant level of differences (p-values) of coefficient c (intercept) (a), coefficient b (linear term) (b), and coefficient a (quadratic term) (c) at each pair of clutter level.

length of NAVA,

$$Y_{ij} = \alpha_j + \beta_j X_{ij} + \delta_j X_{ij}^2 + \epsilon_{ij} \tag{2.2}$$

where $i = 1, 2, \ldots, 24200$ and $j = 1, 2, 3, 4$ standing for $n_c = 25, 50, 75, 100$, respec-

tively; Y_{ij} is the i^{th} traversal length of NAVA at j^{th} group, X_{ij} is the value of d at i^{th} treatment within j^{th} group, and ϵ_{ij} is an error term. The index j denotes the group membership (i.e., clutter level). The coefficient estimates in Model (2.2) are presented in Table 2.3. Table 2.4 suggests that the intercept values of all fitted polynomials in Figure 2.4 are statistically different from each other. The linear and quadratic trends for mean traversal length of clutter levels 25 and 50 are quite similar (see Table 2.4(b) and Table 2.4(c)), whereas the other pairs significantly different.

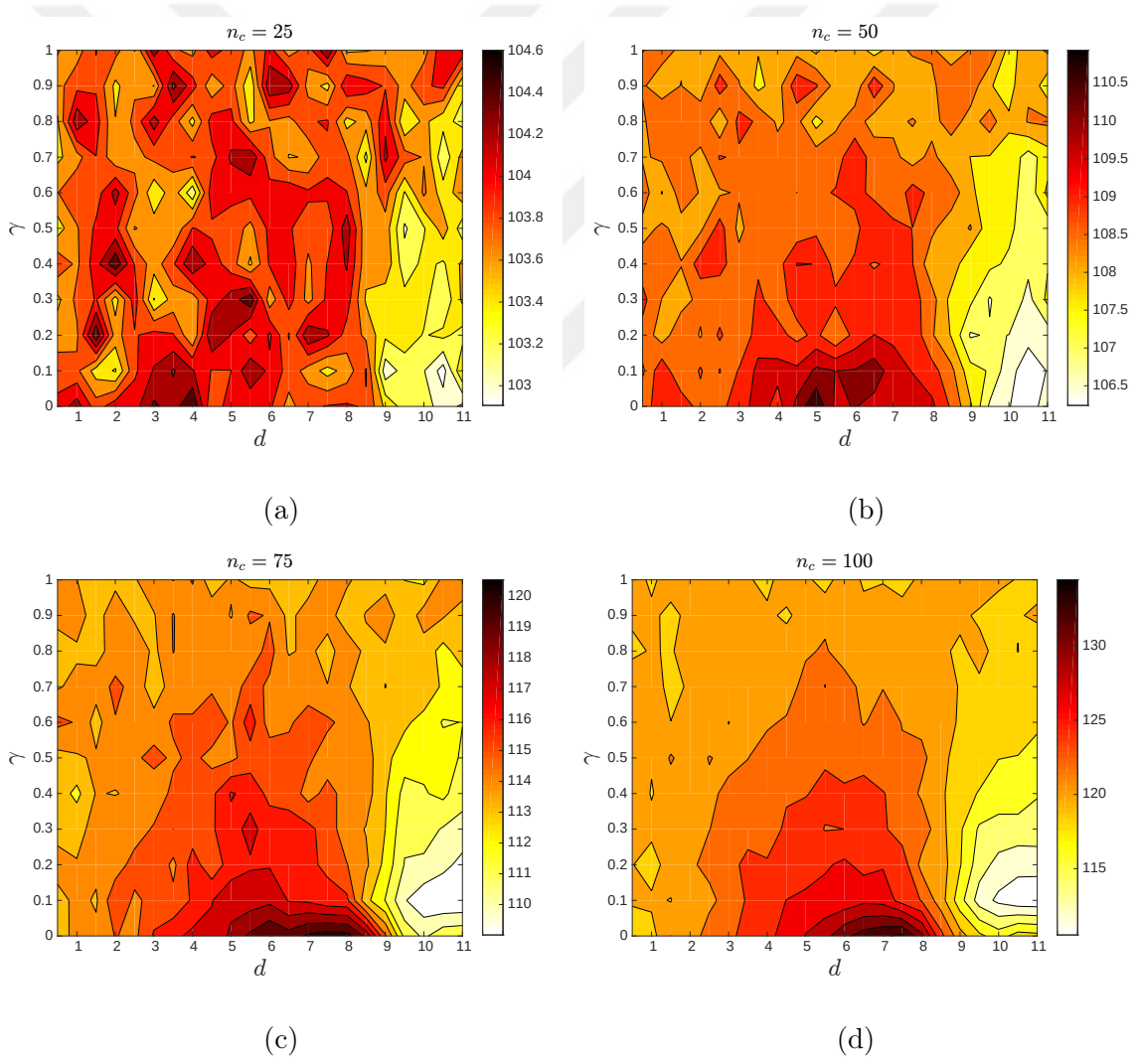


Figure 2.5: Contour plots of mean traversal length versus γ and d values for $n_c = 25, 50, 75, 100$.

We also consider the contour plot of the mean traversal lengths where the impacts of both the γ and d values can be seen simultaneously (see Figure 2.5). The traversal lengths are within a small range, especially for $n_c = 25, 50$ (see Figures 2.5(a-b)). However, for $n_c = 75, 100$, there is much wider range for traversal length and also a trend can be observed in these cases (see Figures 2.5(c-d)). Empirically the mean traversal length is maximized when γ is less than 0.1 and when d is around 7 (from 6 to 8). On the average, for moderate values of d and the smaller values of γ we obtain longer traversal lengths. Also observe that for large values of d, γ values have almost no influence on the traversal length.

In summary, if the whole window is going to be inserted only with the false obstacles with capability of varying the parameters of regularity (i.e., d and γ values), then on the average, traversal lengths tend to increase as the clutter pattern changes from uniformness to regularity. The γ values (esp. 0.1-1.0) have only mild influence on the traversal length, while d (that is especially large for d values between 6-8) has a substantial influence. Considering them together, from OPA's perspective we recommend choosing moderate values of d (within 6 to 8) and smaller γ values ($\gamma < 0.1$) in this setting so as to maximize the total traversal length of NAVA. We also observe that the value d is closely related with the radius r of disk-shaped obstacle. From OPA's perspective the case $d > 2r$ is not desirable since for $d > 2r$ obstacles become too sparse in the environment, whereas the optimal value of d takes place when the ratio d/r is within the interval $(1.33, 1.77)$ together with $\gamma = 0$.

2.3.2 True Obstacles Only Case

The simulation setting is as in the false obstacle only case in Section 2.3.1, with n_c being replaced by n_o. We repeat the same analysis done in Section 2.3.1, but emphasize on the main differences here. For the same levels of number of obstacles, γ and d values, the mean traversal lengths with the true obstacles tend to be longer compared to the false obstacles. Using the model as in the Equation (2.1), we observe that on the average the traversal length tends to increase as the true obstacle pattern

changes from uniformness to regularity provided that $d < 2r$. The trends in traversal length as γ increases is similar to the clutter only case (Figure 2.2), and the slopes and intercept estimates (Table 2.5) are significantly different from each other (Tables 2.6).

	$n_o = 25$	$n_o = 50$	$n_o = 75$	$n_o = 100$
intercept	104.06	111.78	137.63	204.21
slope	0	0.03	2.06	0.93

Table 2.5: Intercept and slope values of regression lines for each categorical group n_o.

intercept

	$n_o = 50$	$n_o = 75$	$n_o = 100$
$n_o = 25$	<0.001	<0.001	<0.001
$n_o = 50$		<0.001	<0.001
$n_o = 75$			<0.001

(a)

slope

	$n_o = 50$	$n_o = 75$	$n_o = 100$
$n_o = 25$	0.812	<0.001	<0.001
$n_o = 50$		<0.001	<0.001
$n_o = 75$			<0.001

(b)

Table 2.6: Significant level of differences (p-values) in intercept (a) and slope (b) for each pair of groups.

When n_o equals 25 or 50, there still remains some obstacle-free spaces in the environment where NAVA can traverse even without any disambiguation. So, the γ values have almost no effect on the traversal length. When $n_o = 75$, the small values of γ becomes more important. When obstacles are distributed more regularly, NAVA starts to take risks (due to the greedy nature of the RD algorithm) rather than circumnavigate the zero risk route. When $n_o = 100$, then the γ value again loses its importance because we already have 100 obstacles which saturates the area regardless of γ values. The mean traversal lengths versus d values are presented in Figure 2.6.

Note that the trends are concave down and similar to the clutter only case (see Figure 2.3). By using a model as in Equation (2.2) we observe that fitted polynomials for

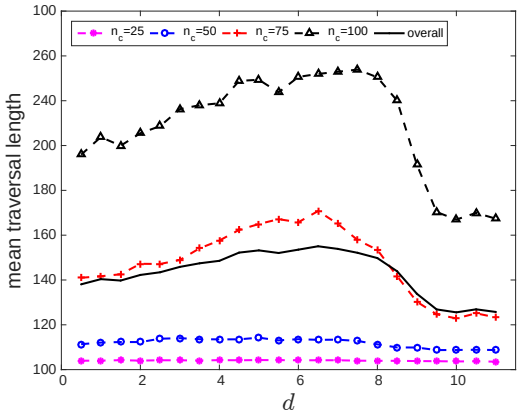
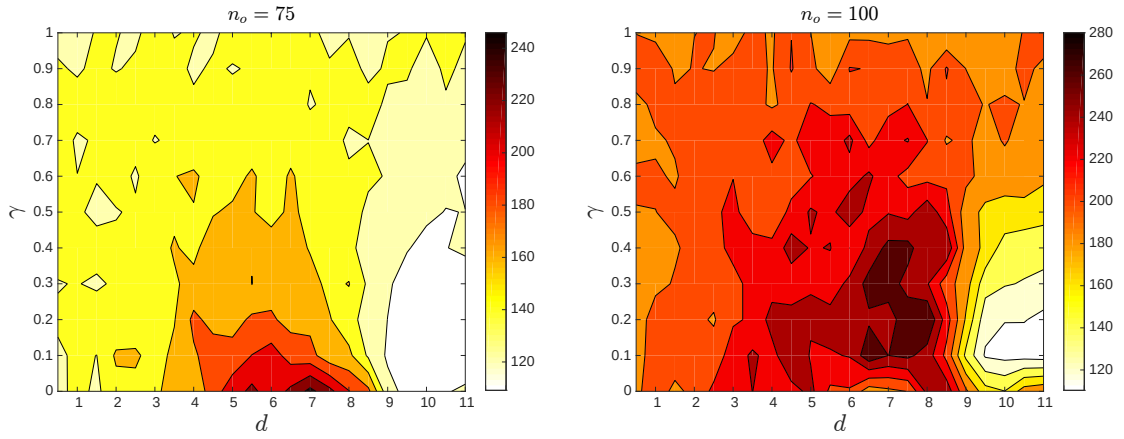


Figure 2.6: Mean traversal length versus γ values for groups $n_o = 25, 50, 75, 100$ together with overall plot (i.e., the average of all groups combined).

true obstacle levels 25 and 50 are similar (hence have similar trends), but the others are significantly different from each other.

We also present the contour plot of the mean traversal lengths versus γ and d values in Figure 2.7 for $n_o = 75, 100$ only since for $n_o = 25, 50$ traversal length weakly depends on d and γ values (hence are omitted). But, when $n_o = 75$ mean traversal length is maximized for d values around 7 together with $\gamma = 0$. As for obstacle number level $n_o = 100$, mean traversal length tends to have higher value for γ values between 0.1 and 0.4, and for d values from 6 to 8.

In summary, if the whole window is to be inserted only with true obstacles with capability of varying the regularity parameters (i.e., d and γ values), then on the average traversal length tends to increase as true obstacle pattern changes from uniformness to regularity. From OPA's perspective, considering γ and d values together, we recommend choosing moderate values of d (around 6 to 8) and smaller γ values (especially, $0.1 < \gamma < 0.4$).



(a) (b)

Figure 2.7: Contour plots of mean traversal length versus γ and d values for $n_o = 75, 100$.

2.3.3 True and False Obstacles Together

In this section we consider the setting in which OPA has the capability of inserting both true and false obstacles simultaneously. The obstacle pattern changes from uniformness to regularity for $n_c = 25, 50, 75$ and $n_o = 25, 50, 75$ such that $n_c + n_o = 50, 75, 100$. An example for the realization of this setting is given in Figure 2.8. From our simulations (see Sections 2.3.1 and 2.3.2), we observe that true obstacles make the traversal length much longer compared to false obstacles when the number of true and false obstacles are same. Hence, from OPA's perspective it is more desirable to insert true obstacles rather than false obstacles (assuming OPA has sufficient amount of both true and false obstacles). For the sake of simplicity, we only present the case where $n_c = 25$ and $n_o = 25, 50, 75$ such that $n_c + n_o = 50, 75, 100$. The trend for traversal length versus γ values for $n_o = 25, 50, 75$ with fixed $n_c = 25$ false obstacles is similar to the one in Figure 2.2. When the whole pattern is uniform (i.e., $\gamma = 1$) the average traversal lengths are smaller and when the pattern is completely regular (i.e., $\gamma = 0$) the average traversal lengths are larger. Based on our analysis, we

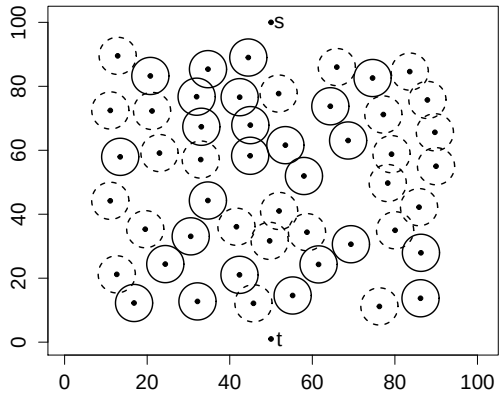
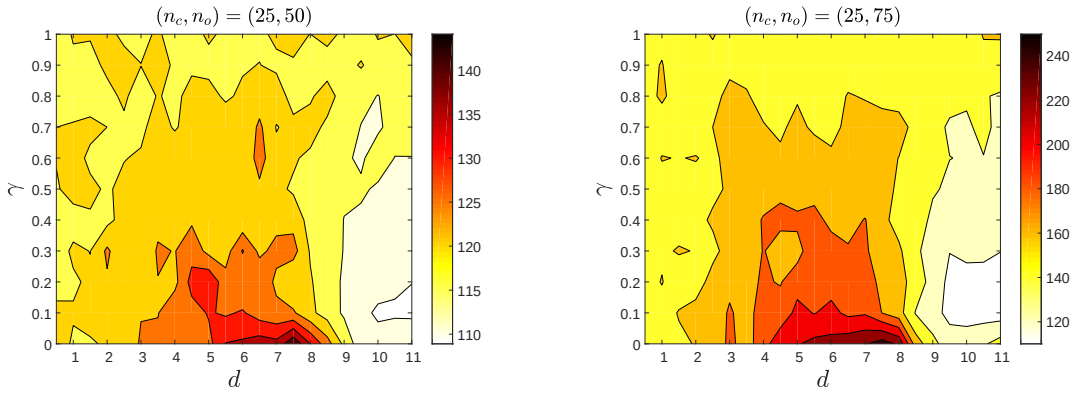


Figure 2.8: The pattern is from the Strauss($n_c + n_o = 50, d = 9, \gamma = 0$) process where dashed circles ($n_c = 25$ with centers in X_C) are the 25 false obstacles and solid circles ($n_o = 25$ with centers in X_O) are the 25 true obstacles.

find no significant difference between the true obstacle number levels 25 and 50 (p-value=0.190), but the trend for true obstacle number level 75 is totally different from the others.

Regarding the influence of d, for obstacle number levels 25, 50, and 75 we still have a concave down trend similar to Figure 2.3 for mean traversal length as d values increase. The mean traversal length is maximized when d is from 5.5 to 6.5.

We also present the contour plot of the mean traversal length versus γ and d values in Figure 2.9 for $(n_c, n_o) = (25, 50)$ and $(n_c, n_o) = (25, 75)$. In the case $(n_c, n_o) = (25, 25)$, traversal length seems to be not affected by d and γ values (hence not presented). Notice that on the average the traversal lengths tend to have larger values when γ values are less than 0.1 and when d values are from 6 to 8 units. For the cases when the number of false obstacles are fixed 50 and 75, we obtain very similar results as the case when the false obstacle number level is fixed to 25. Hence, the plots for $n_c = 50$ with $n_o = 25, 50, 75$ and $n_c = 75$ with $n_o = 25, 50, 75$ cases are not presented.



(a) (b)

Figure 2.9: Contour plots of the mean traversal length versus γ and d values when (a) $(n_c, n_o) = (25, 50)$, (b) $(n_c, n_o) = (25, 75)$.

In summary, if the whole window is to be inserted with both false and true obstacles together where the overall pattern changes from uniformness to regularity, then on the average traversal lengths tend to increase. Based on our observation, in order to maximize NAVA's traversal length, we suggest choosing small γ values (less than 0.1) and moderate d values (from 6 to 8) for OPA to insert obstacles in the environment. If feasible, we also recommend OPA insert more true obstacles than false obstacles, and mix them so as to trick NAVA to run into more true obstacles and hence do more disambiguations and replan its route to a longer path.

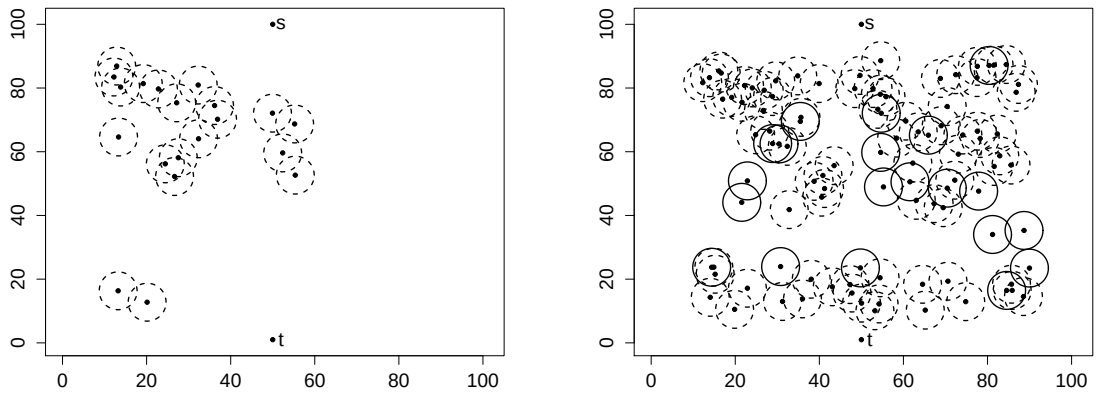
2.4 Obstacle Pattern Ranging from Uniformness to Clustering

In this section, we consider the case that OPA inserts obstacles from patterns changing from uniformness to clustering. To investigate this scenario we use the Matérn clustering point process denoted Matérn(κ, r_0, μ) wherein a given window, Ω, first we uniformly generate 'parent points', and then for each parent point we generate 'offspring points', independently and uniformly distributed in a ball of radius r_0 cen-

tered around the parent. In the Matérn(κ, r_0, μ) process, κ is the intensity of 'parent' points from Poisson process, each parent has a Poisson(μ) number of offsprings independently and uniformly distributed in a disc of radius r_0 centered around the parent (Baddeley et al., 2008). Then parents are discarded from the region, and only offspring points are kept. Throughout this section the average number of points (i.e., offsprings μ) per parent point is taken to be 10, and the radius of a ball (r_0) around the parent varies from 5 to 50 with an increment size of 5 units. Notice that as the radius gets larger, the pattern gets closer to uniformness.

In Aksakalli and Ceyhan (2012), the clustering pattern Matérn(κ, r_0, μ), was employed with parameters $\kappa = 10, r_0 = 10$, and $\mu = 10$ so that on the average there are $\kappa \times \mu = 100$ clutter points. And, the number of true obstacles n_o was taken to be 20, 30, 40, 50, and 60. They observed that NAVA tends to have lower traversal length for clustered point pattern. In the current study, we want to address how the traversal length changes when both κ (hence μ) and r_0 change under clustering by considering various values of κ and r_0 simultaneously. We investigate three different cases. In line with Section 2.1, the whole window is inserted with clutter disks (i.e., false obstacles) only or true obstacles only, then the whole window is inserted with both true and false obstacles together (see Figure 2.10). The number of parent points κ is chosen to be 2, 4, 6 and 8 and for each parent there are on the average $\mu = 10$ offsprings.

We present the mean traversal length versus clustering radius for the false obstacle only and true obstacle only cases in Figure 2.11. If there are only false obstacles, then as the pattern changes from uniformness to clustering, the average traversal length decreases, especially for small values of cluster radius. For large values of cluster radius the trend is almost same as in the uniformness pattern. Similarly, if there are only true obstacles, as the pattern changes from uniformness to clustering, the average traversal length decreases as the clustering radius decreases; and the rate of decrease is remarkably sharp for smaller radius values. For the case when there are both false and true obstacles, we investigate the situations where the number of false

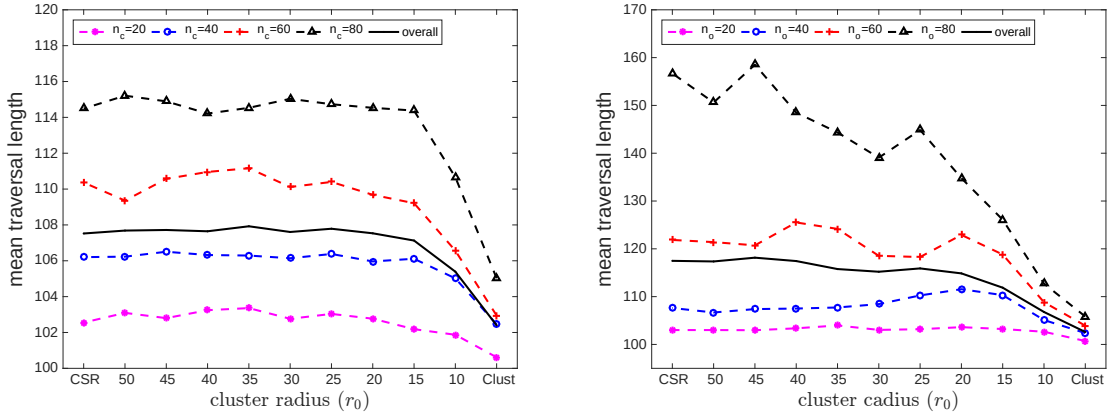


(a) (b)

Figure 2.10: (a) A realisation of clutter disks from the Matérn($\kappa = 2, r_0 = 15, \mu = 10$) process shown in dashed circles. (b) The pattern is from the Matérn(8,15,10) process where dashed circles indicate the false obstacles ($n_c = 60$) and solid circles indicate the true obstacles ($n_o = 20$).

obstacles is fixed to $n_c = 20$ and the number of true obstacles n_o takes values $20, 40,$ and 60. That is to say, we consider the pairs $(n_c, n_o) = (20, 20), (20, 40)$, and $(20, 60)$. The trend is similar to the cases in Figure 2.11 (hence not presented).

Based on our Monte Carlo simulation results, we observe that on the average traversal length tends to decrease for true obstacles, false obstacle and both as the clustering radius decreases and at a sharper rate for smaller radius values. When the pattern gets more clustered, obstacles accumulate around one or multiple points (i.e., cluster centers) and as a result it is more likely to have more empty spaces inside the window with a shorter length from s to t and this enables NAVA to attain a quick traversal with encountering few or no obstacles.



(a) (b)

Figure 2.11: Mean traversal length versus cluster radius values when there are (a) only clutter disks with $n_c = 20, 40, 60, 80$ together with overall plot (i.e., the average of all groups combined), (b) only true obstacles with $n_o = 20, 40, 60, 80$ together with overall plot (i.e., the average of all groups combined). Notice that vertical axes are differently scaled.

2.5 Stochastic Ordering of Traversal Lengths

We next consider the stochastic ordering of traversal lengths under various obstacle patterns. Recall that NAVA is designed to traverse from source s to target t on an (s,t) walk, with each edge weighted by the Euclidean distance and the disambiguation cost of an obstacle. Given n (i.e., $n_o + n_c$), let $\rho = n_o/n_c$ be the ratio of true obstacles to false obstacles. Then, for any fixed ρ value, suppose that there are m many distinct distances (avoiding loops) with i^{th} distance occurring k_i times. So, denote such an (s,t) path as π_{ij}, $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, k_i$ so that there are in total $M = \sum_{i=1}^{m} k_i$ many (s,t) paths on the usual integer grid. When there are no obstacles ($n = 0$) in the environment, let ℓ_{ij} be the corresponding Euclidean length for path π_{ij} and without loss of generality we can assume that $\ell_{1j} < \ell_{2j} < \ldots < \ell_{mj}$. Notice that for any fixed i, there are k_i paths of equal length, i.e., $\ell_{ij} = \ell_{ij'}$ for any

$j, j' = 1, 2, \ldots, k_i$.

Now, let L_{ij} be the traversal length for path π_{ij} in the presence of true or false obstacles. In the absence of obstacles ($n = 0$) we have the equality $L_{ij} = \ell_{ij}$ for all i, j. When there are only false obstacles ($n > 0$ and $\rho = 0$, i.e., $n_o = 0$), then we need to update the value of L_{ij} by using the weight function defined by the RD algorithm. Recall that, when there are only clutter disks the set $X = X_C$ where X_C is the set of points of centers of disk shaped obstacles $D_r(x)$ (a disc of radius r centered at a point x) for $x \in X_C$ with fixed radius $r > 0$. And, for $x \in X$, $p(x)$ is the probability that $D_r(x)$ is a true obstacle with $c(x)$ being the disambiguation cost of $D_r(x)$. In our case, the disambiguation cost $c(x) = c > 0$ is fixed. If we ignore the probabilities, we have

$$L_{ij} = \ell_{ij} + w_{ij} \times c \quad \text{with} \quad w_{ij} = \sum_{x \in X_C} \mathbf{1}_{D_r(x) \cap \pi_{ij} \neq \emptyset}$$

where w_{ij} gives the number of obstacles intersecting the edges of path π_{ij}. On the other hand, if we do consider the probabilities of disk shaped obstacle being true obstacle, then we update each edge of path π_{ij}. Namely, if $e \in \pi_{ij}$ then,

$$w(e) = \ell(e) + \frac{1}{2} \times F(e) \quad \text{with} \quad F(e) = \sum_{x \in X_C} \mathbf{1}_{\{D_r(x) \cap e \neq \emptyset\}} \times \left(\frac{c}{1 - p(x)} \right)$$

So that $L_{ij} = \sum_{e \in \pi_{ij}} w(e)$.

Notice that in the updated version for any fixed i, $L_{ij} \neq L_{ij'}$ in general, because the number of obstacles intersecting the paths of same Euclidean length π_{ij} and $\pi_{ij'}$ may not be equal. If we consider all L_{ij} values in the clutter only case in an array (1 by M) with M many entries so that $L_{(k)}$ is the k^{th} smallest length, then RD algorithm choses $L_{(k)}$ over $L_{(k')}$ for $k < k'$.

Let L_{ij}^S and L_{ij}^U be the traversal lengths of path π_{ij} under regularity ($\gamma = 0$) and uniformness ($\gamma = 1$), respectively. Under regularity, the path π_{ij} is more likely to intersect an obstacle compared to uniformness. So, it is more likely that $L_{ij}^U \leq_{st} L_{ij}^S$, i.e., the traversal length is stochastically smaller under the uniformness compared to that under regularity. Indeed, one can extend the above argument to the case

that at fixed d, let L_{ij}^{γ} denote the traversal length for Strauss(n, d, γ) pattern. Then, $L_{ij}^{\gamma} \geq_{st} L_{ij}^{\gamma'}$ whenever $\gamma < \gamma'$. Similarly, $L_{ij}^{r_0}$ be the traversal length for Matérn(κ, r_0, μ) pattern and let L_{ij}^{M} denote the traversal length of path π_{ij} under clustering ($r_0 \approx 0$) for fixed κ, μ. Then for $r_0 < r_0'$, $L_{ij}^{r_0} \leq_{st} L_{ij}^{r_0'}$ since under Matérn(κ, r_0, μ) pattern any π_{ij} is less likely to hit an obstacle compared to that under Matérn(κ, r_0', μ) whenever $r_0 < r_0'$. So, we have proved the following:

Proposition 2.5.1 *: Whenever $X = X_C$ (i.e.,OPA only inserts false obstacles to our standard window $[0, 100] \times [0, 100]$), we have*

(i) $L_{ij}^{M} \leq_{st} L_{ij}^{U} \leq_{st} L_{ij}^{S}$ *(defined as above).*

(ii) $L_{ij}^{\gamma} \geq_{st} L_{ij}^{\gamma'}$ *for* $\gamma < \gamma'$.

(iii) $L_{ij}^{r_0} \leq_{st} L_{ij}^{r_0'}$ *for* $r_0 < r_0'$.

In the true obstacle only case ($n > 0$ and $\rho = \infty$, i.e., $n_c = 0$), if an obstacle intersects a path then it renders that path non-traversable. So, if a shorter path hits a true obstacle, RD resets the algorithm whenever NAVA hits the obstacle and then picks another longer path. That is to say, when we have true obstacles the total number of distinct traversable paths reduces compared to the clutter only case. Hence, traversal length in the true obstacle case tends to be larger compared to that in the false obstacle only case. Furthermore, by the same reason as above the same stochastic orderings occur under regularity, clustering and uniformness patterns.

Proposition 2.5.2 *: Whenever $X = X_O$ (i.e.,OPA only inserts true obstacles to our standard window $[0, 100] \times [0, 100]$), we have*

(i) $L_{ij}^{M} \leq_{st} L_{ij}^{U} \leq_{st} L_{ij}^{S}$

(ii) $L_{ij}^{\gamma} \geq_{st} L_{ij}^{\gamma'}$ *for* $\gamma < \gamma'$.

(iii) $L_{ij}^{r_0} \leq_{st} L_{ij}^{r_0'}$ *for* $r_0 < r_0'$.

One can also write stochastic ordering for clutter only and true obstacle only cases as follows. Let L_{ij}^C be the traversal length of path π_{ij} under the clutter only case and L_{ij}^O be the traversal length of path π_{ij} under the true obstacle only case. Then with the same obstacle pattern of $n_c = n$ in the clutter only case and $n_o = n$ in the true obstacle only case we have,

Proposition 2.5.3 *: With the same obstacle pattern and the same number of obstacles,*

$$L_{ij}^C \leq_{st} L_{ij}^O.$$

This holds because if true obstacles were also clutter, then we would have $L_{ij}^C \stackrel{d}{=} L_{ij}^O$. But, when NAVA runs into a true obstacle, then it has to stop and choose a longer traversable path while if the obstacle was false NAVA would traverse through it. Since the cost of disambiguation is same, we have a reduction in the number of traversable paths under true obstacle only case with L_{ij}^C values being more likely to be smaller than L_{ij}^O values. Let $L_{ij}^{C,O}$ be the traversal length of path π_{ij} under the clutter & true obstacle cases.

Corollary 2.5.3.1 *: With the same obstacle pattern and the same number of obstacles,*

$$L_{ij}^C \leq_{st} L_{ij}^{C,O} \leq_{st} L_{ij}^O.$$

The result also extends to the mixed obstacle case. One can also show that with everything else being same, and $\rho = n_o/n_c$ and keeping the number of clutter equal, we have $L_{ij}^{\rho} \leq_{st} L_{ij}^{\rho'}$ whenever $\rho < \rho'$ (but, notice that this result does not extend if the number of clutter values are different). Indeed, this case can be generalized whenever total number of obstacles is fixed (we will be studied in detail in Section 2.6).

Proposition 2.5.4 *: Whenever $X = X_C \cup X_O$, and n_c is fixed, then for $\rho < \rho'$*

(i) $L_{ij}^M \leq_{st} L_{ij}^U \leq_{st} L_{ij}^S$

(ii) $L_{ij}^{\gamma} \geq_{st} L_{ij}^{\gamma'}$ *for* $\gamma < \gamma'$.

(iii) $L_{ij}^{r_0} \leq_{st} L_{ij}^{r_0'}$ *for* $r_0 < r_0'$.

To conclude, at any path π_{ij}, the cost under regularity would tend to be higher compared to uniformness. The main reason for this is that under regularity, it is more likely for an obstacle to be placed on an edge on π_{ij} compared to uniformness, since under regularity obstacles will be more evenly distributed compared to uniformness. On the other hand, for any path π_{ij} the cost under clustering would tend to be lower compared to uniformness. Because, under clustering the obstacle is less likely to hit an edge on π_{ij} compared to uniformness. Since, under clustering obstacles will be accumulated around parents or hotspots and thus form clumps so that π_{ij} has more chance of not falling inside these clumps compared to uniformness. Thus, if the obstacle pattern is regular then the probability of maximizing the mean traversal length of NAVA is higher than any other cases.

2.6 Mean Traversal Length Versus Ratio of Number of True to False Obstacles

So far we denote the number of clutter (false obstacles) by n_c and the number of true obstacles by n_o, and $n = n_c + n_o$ and we compute the traversal lengths for each n_c, n_o combinations. In this section, we investigate the trends for mean traversal length with respect to the ratio $\rho = n_o/n_c$ when the sum $n_o + n_c$ is fixed. In our experimental setting the total number of obstacles to be inserted into the study window is equal to 100. So, the case $n_o = 0$ corresponds to deploying the study window only with clutter, and $n_c = 0$ means inserting only true obstacles. To measure the traversal length for intermediate values of pairs of (n_o, n_c), we introduce a new variable $\rho = n_o/n_c$ taking the values $\{0, 1/4, 1/2, 1, 2, 4, \infty\}$. Thus, extreme cases of $\rho = 0$ is equivalent to the case $n_o = 0$ and $\rho = \infty$ is equivalent to the case $n_c = 0$.

In the following subsections, we investigate the case where the obstacle pattern changes from the complete spatial uniformness (uniformness) to regularity. We

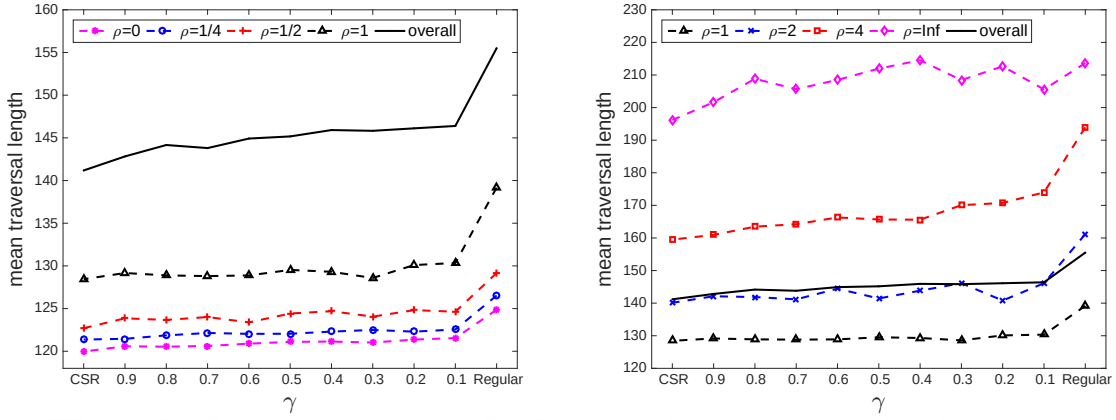
use 11×11 partition of $(\gamma, d) \in (0,1) \times (1,11)$ (i.e., γ and d are parameters of Strauss(n, d, γ)), so our current investigation is much more detailed for γ compared to Aksakalli and Ceyhan (2012), and investigation for d is different from that in Aksakalli and Ceyhan (2012).

We also investigate the case where the obstacle pattern changes from the uniformness to clustering. In our experimental setting, we choose the κ values $\{1, 2, 3, ..., 10, 11\}$ and the radius r_0 around the parent varies from 5 to 50 units with an increment size of 5 units (i.e., κ and r_0 are parameters of Matérn(κ, r_0, μ)). As the radius gets larger, the pattern formed is getting closer to uniformness. In Aksakalli and Ceyhan (2012), the cluster radius r_0 and the average offspring per parent μ were both taken to be 10. In our study, we increase the range of r_0 and κ values.

2.6.1 Obstacle Pattern Ranging from Uniformness to Regularity

Our goal is to determine the trend for the mean traversal length of NAVA as a function of the level of regularity. In our experimental setting, there are 11 different values of d $(1, 2, ..., 10, 11)$, 11 different values of γ $(0, 0.1, ..., 0.9, 1)$, 7 different values of ρ (0, 1/4, 1/2, 1, 2, 4, inf), and 100 Monte Carlo simulations for each of these combinations. First, we only consider the influence of γ values, and then we will discuss the effect of d values. We model the traversal length as in Equation (2.1) where the categorical level will be ρ. Since there are 7 categorical groups, for simplicity we split the corresponding interaction plots into two parts. In Figure 2.12(a) we show the interaction plot of mean traversal length versus γ values for groups $\rho = 0, 1/4, 1/2, 1$ and the overall plot (i.e., the average of all 7 distinct ρ groups). And, in Figure 2.12(b) we show the interaction plot of mean traversal length versus γ values for groups $\rho = 1, 2, 4, \infty$ and the overall plot.

We observe that the intercept values for $\rho = 1/4, 1/2$ are not significantly different from $\rho = 0$ (p-values 0.4071, 0.0524, respectively). And, the slopes of regression lines for $\rho = 1/4, 1/2, 1$ are not significantly different than the slope of group $\rho = 0$ (p-values 0.9817, 0.7207, 0.1984, respectively). Moreover, the slope values for all groups

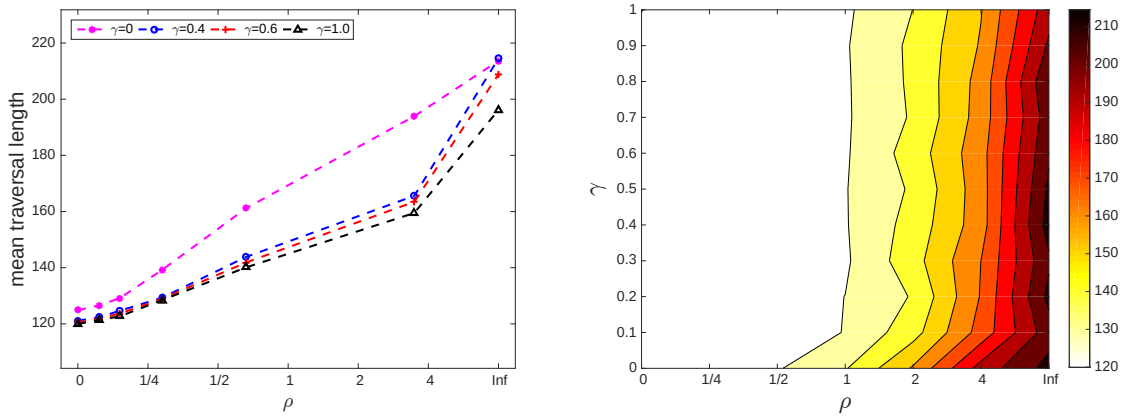


(a) (b)

Figure 2.12: Mean traversal length versus γ values for groups (a) $\rho = 0, 1/4, 1/2, 1$ together with overall plot (the average of all 7 distinct ρ groups), (b) $\rho = 1, 2, 4, \infty$ together with overall plot (the average of all 7 distinct ρ groups). Notice that vertical axes are differently scaled.

are positive (Table 2.7), which means that the mean traversal length tends to increase as the gamma (γ) values decrease. In all groups, the average traversal length is maximized when $\gamma = 0$ (i.e., when the pattern is completely regular) and the mean traversal length tends to increase as ρ gets larger (i.e., intercept value increases as ρ increases). It is also interesting that the overall trend of mean traversal length is almost the same as the case $\rho = 2$ (see Figure 2.12(b)).

Moreover, in Figure 2.13 we show the plot of mean traversal length versus ρ values for various choices of γ and the contour plot of mean traversal length versus γ and ρ values. In Figure 2.13(a), the mean traversal length tends to be higher (and thus maximized) when $\gamma = 0$ for each ρ values. And, in Figure 2.13(b) layers of color are formed for each different values of ρ implying that the mean traversal length tends to be larger as ρ value increases. For fixed ρ value, the trend when $\gamma = 0$ is different from other values of γ showing that the traversal length is maximized when $\gamma = 0$. Figure 2.13(b) also summarizes the plots given in Figure 2.12.



(a) (b)

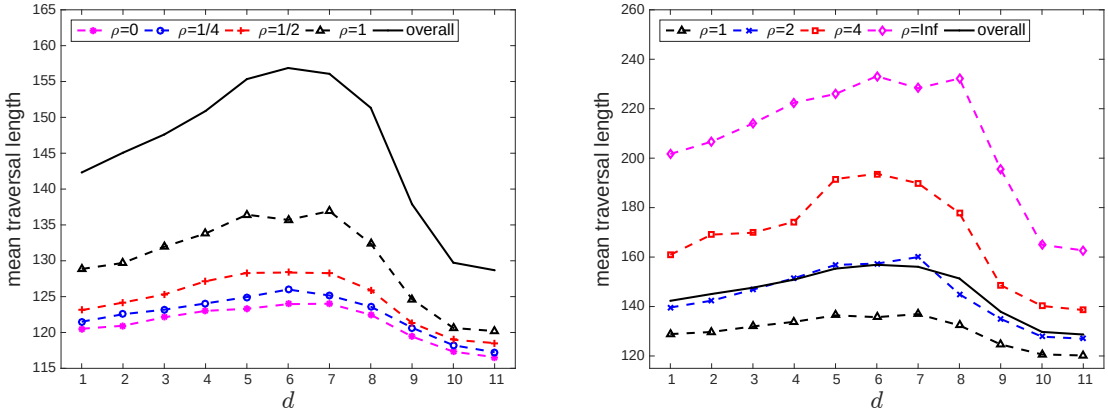
Figure 2.13: (a) Mean traversal length versus ρ values for groups $\gamma = 0, 0.4, 0.6, 1$, (b) average contour plot of mean traversal length versus γ and ρ values.

	$\rho = 0$	$\rho = 1/4$	$\rho = 1/2$	$\rho = 1$	$\rho = 2$	$\rho = 4$	$\rho = \infty$
intercept	119.50	120.69	122.29	126.73	137.52	154.55	201.07
slope	0.29	0.29	0.36	0.56	1.15	2.33	1.14

Table 2.7: Intercept and slope values of regression lines for each categorical group ρ.

We also record the intercept and slope values of regression lines for each categorical group ρ (see Table 2.7). If we repeat the model as in Equation (2.1) for each pair of groups, then we will get the pairwise significance levels of differences in intercept and slope similar to Table 2.2. Thus, we have more information about how exactly each categorical group differs from one another. For instance, Table 2.7 and Figure 2.12 shows that although the intercepts are different for $\rho = 2$ and $\rho = \infty$ groups, the trend for mean traversal length of NAVA is similar since their slopes (1.15 and 1.15, respectively) are not significantly different (p-value=0.974).

Next, we consider the effect of d values on mean traversal length of NAVA for different ρ groups. Similar to Figure 2.3, we will fit the data set into the best quadratic

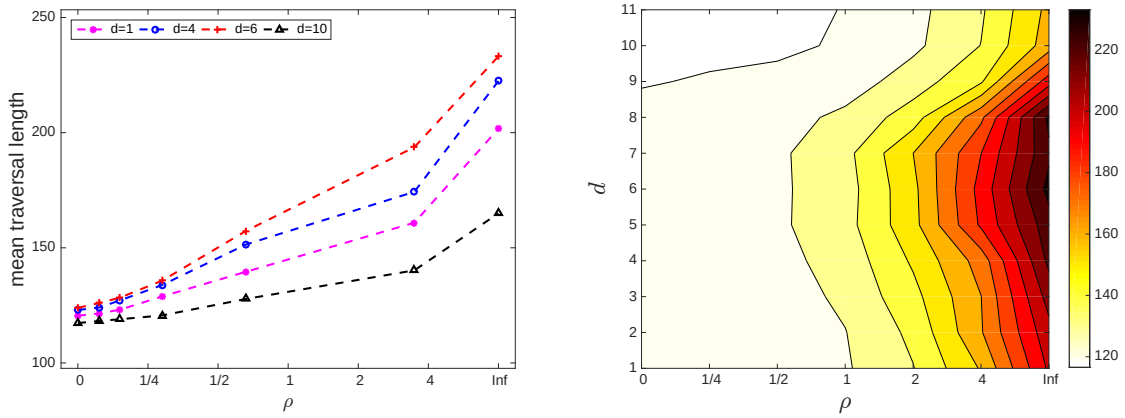


(a) (b)

Figure 2.14: Mean traversal length versus d values for groups (a) $\rho = 0, 1/4, 1/2, 1$ together with overall plot (the average of all 7 distinct ρ groups), (b) $\rho = 1, 2, 4, \infty$ together with overall plot (the average of all 7 distinct ρ groups). Notice that vertical axes are differently scaled.

polynomial and record the corresponding coefficients because the curves in Figure 2.14 suggest a concave down trend. In Figure 2.14, we provide graphs separately for ρ levels $\rho = 0, 1/4, 1/2, 1$ and $\rho = 1, 2, 4, \infty$ together with overall plot (the average of all 7 distinct ρ groups) in order to observe the general trend. Clearly, mean traversal length gets higher, reaches a peak and then decreases as d values increase. And, concave down trend gets more emphasized as the ρ increases.

Moreover, in Figure 2.15 we show the plot of mean traversal length versus ρ values for various choices of d and the contour plot of mean traversal length versus d and ρ values. In Figure 2.15(a), the mean traversal length tends to be higher when $d = 6$ (and similarly between 5 and 8) for each ρ values. Observe that when d gets smaller values (less than 3) or larger values (greater than 9), then the mean traversal length tends to be smaller than the case d is between 5 and 8. In Figure 2.15(b) layers of color are formed for each different values of ρ implying that the mean traversal length tends to be larger as ρ value increases. Figure 2.15(b) also summarizes the



(a) (b)

Figure 2.15: (a) Mean traversal length versus ρ values for groups $d = 1, 4, 6, 10$, (b) average contour plot of mean traversal length versus d and ρ values.

plots (concave down trend) given in Figure 2.14.

We also present the contour plot of the mean traversal lengths versus γ and d values in Figures 2.16 and 2.17. Notice that on the average the traversal length tends to have larger values when γ values are less than 0.1 and when d values are concentrated around 7 for ρ levels $\rho = 0, 1/4, 1/2, 2$. As the ρ values increases, the contour plots get more emphasized and for $\rho = 1$ level the mean traversal length is maximized when γ is less than 0.1 and d is around 5 and 7. In Figure 2.17, for $\rho = 4$ the average traversal length is maximized when γ is less than 0.1 and d is around 5 and 8. Unlike previous cases, for $\rho = \inf$ the mean traversal length tends to have higher values when γ is between 0.1 and 0.4 together with d values between 6 and 8. Based on the overall contour plot of ρ levels (see Figure 2.17(d)), we observe that the NAVA tends to have higher mean traversal length for smaller γ values (i.e., smaller than 0.1) and moderate d values around 7. As a consequence, for fixed number of obstacles $n = n_o + n_c$ the ratio $\rho = n_o/n_c$ is also important in maximizing the traversal length of NAVA. So,

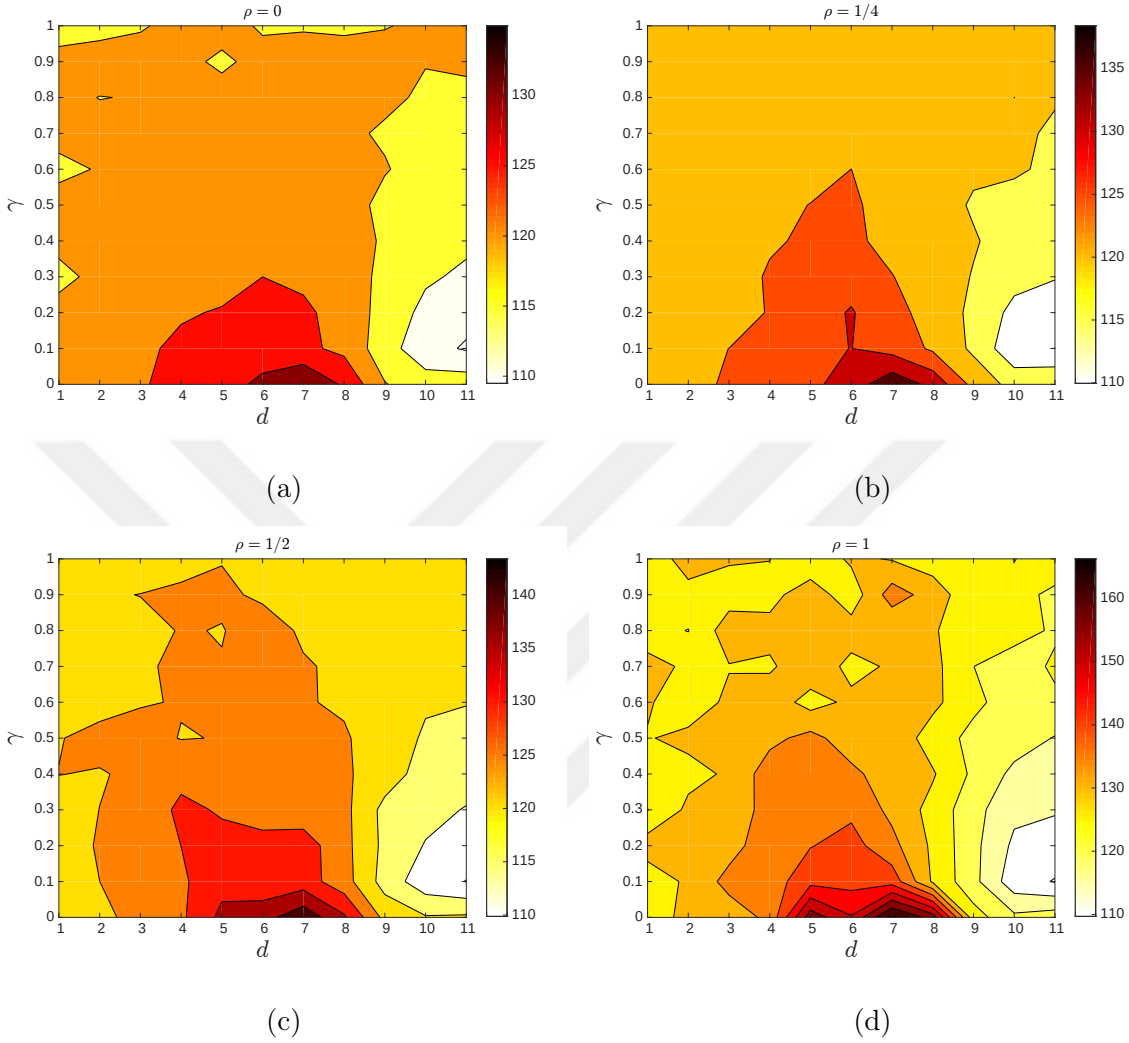


Figure 2.16: Overall average contour plots of the mean traversal length versus γ and d values for (a) $\rho = 0$, (b) $\rho = 1/4$, (c) $\rho = 1/2$, and (d) $\rho = 1$.

Proposition 2.6.1 : *Whenever* $X = X_C \cup X_O$, *and* $n = n_c + n_o$ *is fixed, then* $L_{ij}^{\rho} \leq_{st} L_{ij}^{\rho'}$ *for* $\rho < \rho'$.

Thus, from OPA's perspective considering γ and d values together, we recommend choosing moderate values of d (around 7) and smaller γ values (less than 0.1). And, given the total number of obstacles ($n = n_c + n_o$ is fixed), we recommend to choose ρ value as large as possible so as to maximize the total traversal length of NAVA.

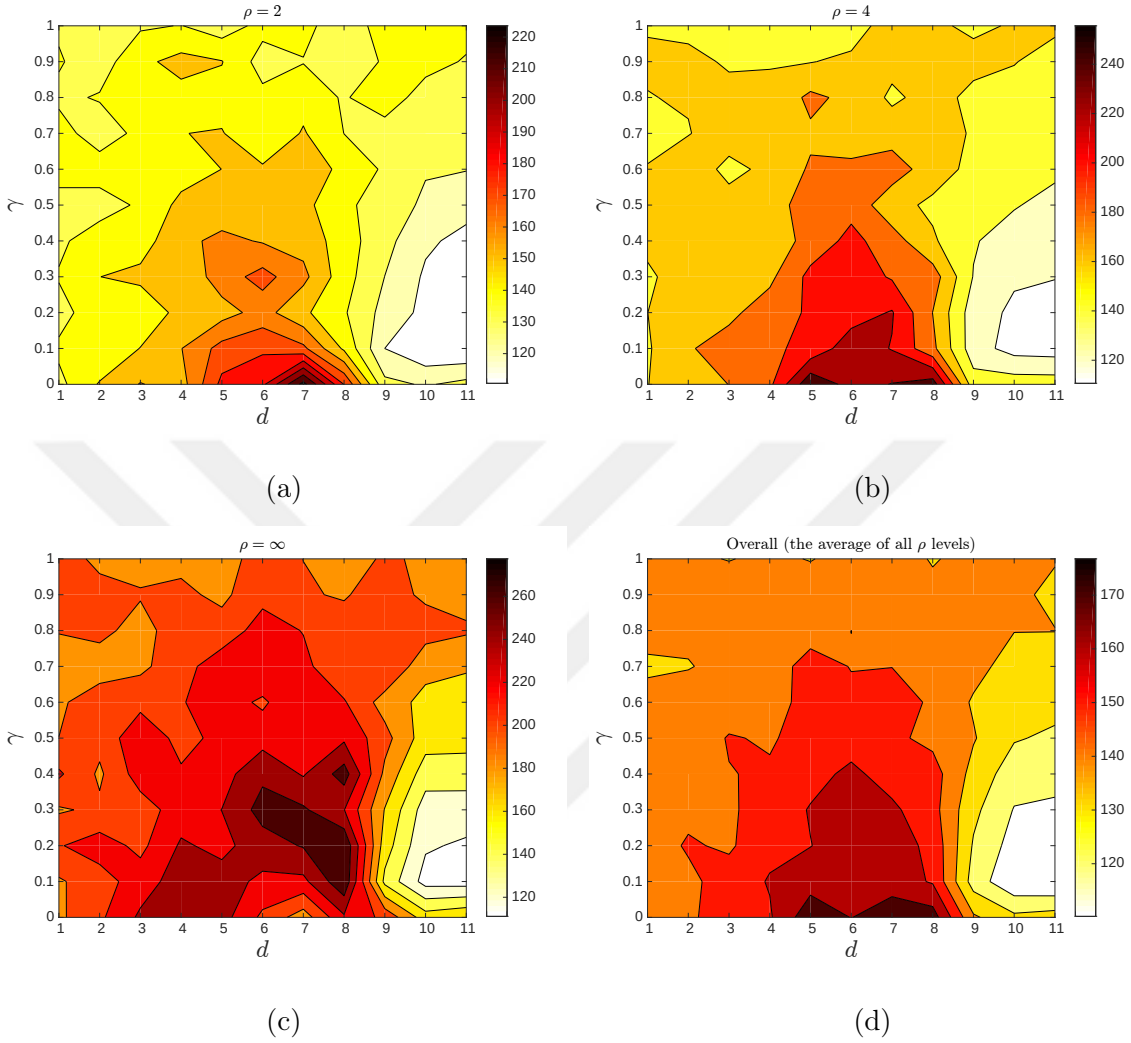


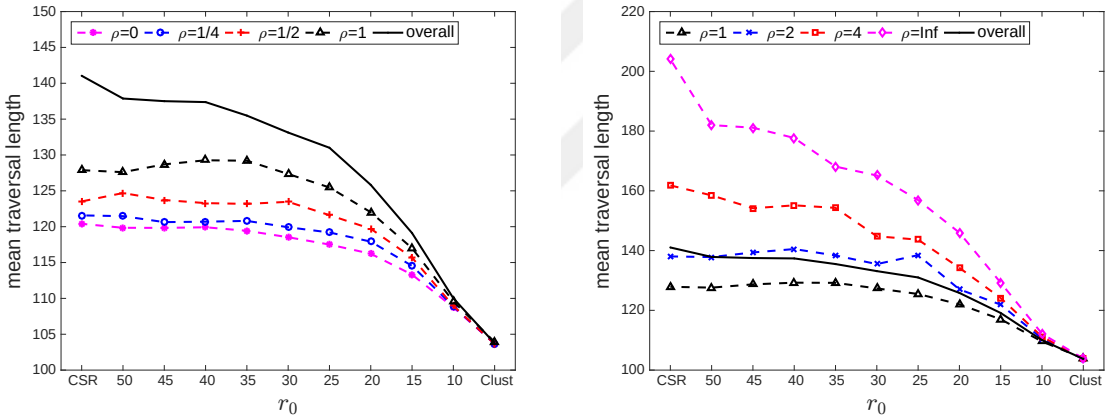
Figure 2.17: Overall average contour plots of the mean traversal length versus γ and d values for (a) $\rho = 2$, (b) $\rho = 4$, (c) $\rho = \infty$, and (d) overall (the average of all ρ levels).

2.6.2 Obstacle Pattern Ranging from Uniformness to Clustering

In this Section, we investigate the trend for mean traversal length of NAVA as a function of the level of clustering. To generate the clustering point pattern we use Matérn(κ, r_0, μ) as in Section 2.4. In our experimental setting, there are 11 different values of κ (i.e., $1, 2, \ldots, 11$), 11 different values of cluster radius r_0 (i.e., $5, 10, \ldots, 50$),

7 different values of ρ (i.e., $0, 1/4, 1/2, 1, 2, 4, \infty$), and 100 Monte Carlo simulations for each these combinations. Since the total number of obstacle is fixed to $n = 100$, so on the average the μ value automatically equals to n/κ.

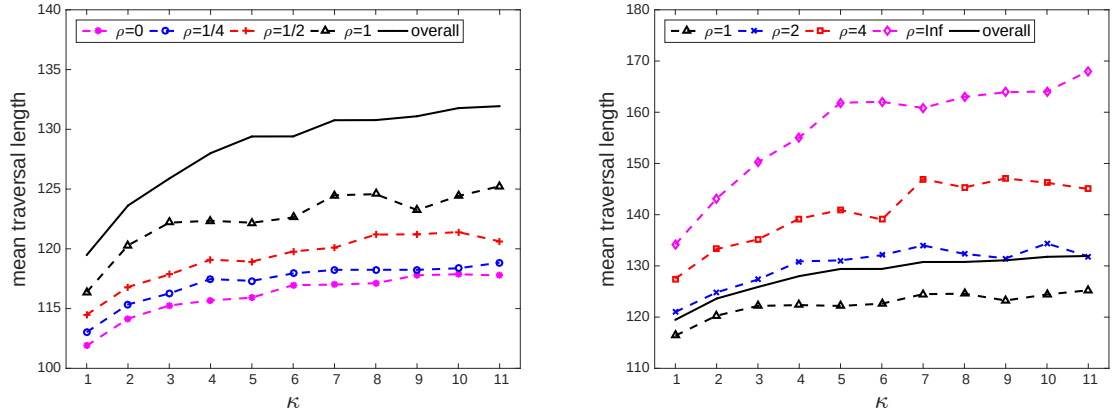
Ignoring κ, in Figure 2.18 we observe that the mean traversal length tends to decrease as the cluster radius r_0 decreases for each ρ level (concave down decrease trend). That is to say, the mean traversal length tends to decrease as the obstacle pattern changes from uniform to clustering. In all groups, the average mean traversal length is maximized when the obstacle pattern is uniform and minimized when the obstacle pattern is clustered. The trends is concave down, and as the ρ value increases the decrease for mean traversal length gets more emphasized.



(a) (b)

Figure 2.18: Mean traversal length versus r_0 values for groups (a) $\rho = 0, 1/4, 1/2, 1$ together with overall plot (the average of all 7 distinct ρ groups), (b) $\rho = 1, 2, 4, \infty$ together with overall plot (the average of all 7 distinct ρ groups). Notice that vertical axes are differently scaled.

Ignoring r_0, in Figure 2.19 the mean traversal length tends to increase as the number of parent points increases for each ρ level (concave down increase trend). But, the increase in trend slows down after $\kappa = 5$ because as the number of parent points increases the obstacles pattern gets less clustered. In both Figures 2.18 and

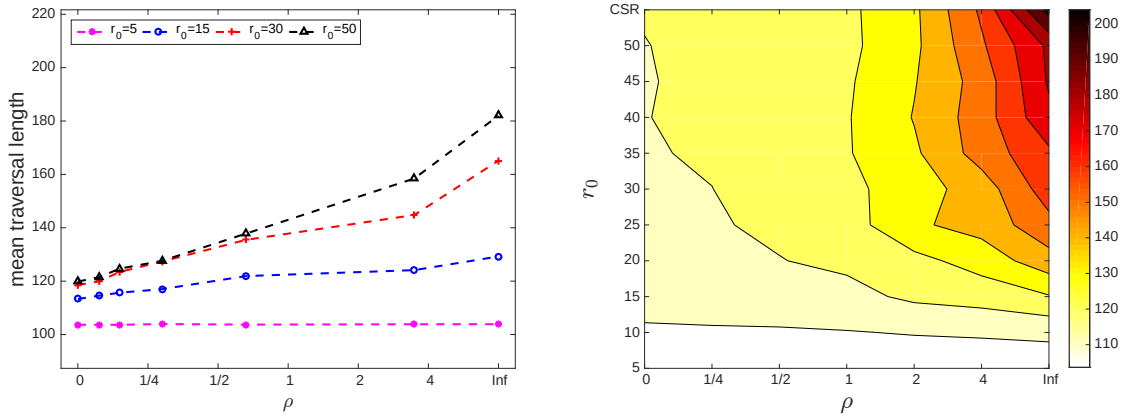


(a) (b)

Figure 2.19: Mean traversal length versus κ values for groups (a) $\rho = 0, 1/4, 1/2, 1$ together with overall plot (the average of all ρ levels), (b) $\rho = 1, 2, 4, \infty$ with overall plot (the average of all ρ levels). Notice that vertical axes are differently scaled.

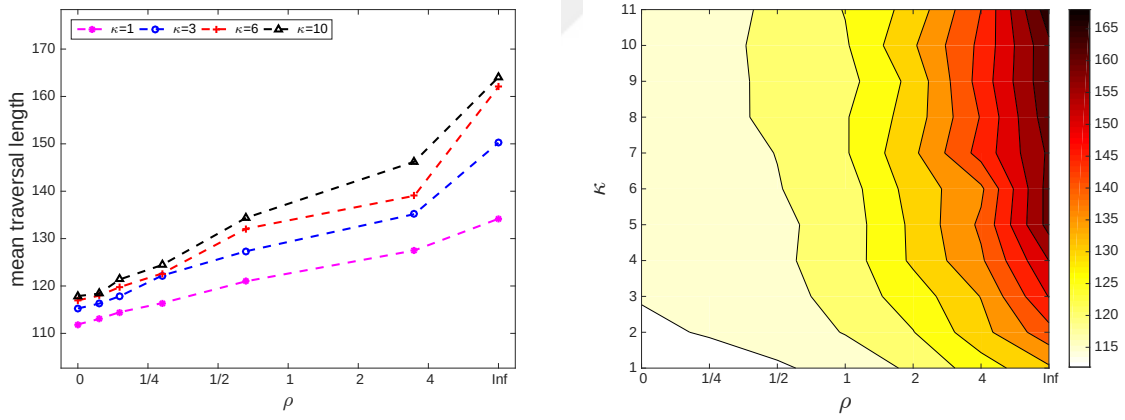
2.19, the overall plot is similar to the group level $\rho = 2$. Moreover, in Figure 2.20 we show the plot of mean traversal length versus ρ values for various choices of r_0 and the contour plot of mean traversal length versus r_0 and ρ values. In Figure 2.20 (a), the mean traversal length tends to be higher when $r_0 = 50$ (i.e., eventually CSR) for each ρ values. In Figure 2.20 (b) layers of color are formed for each different values of ρ implying that the mean traversal length tends to be larger as ρ value increases. Figure 2.20 (b) also summarizes the plots (concave down trend) given in Figure 2.18. Similarly, in Figure 2.21 (a) as the number of parent points (κ) decreases (i.e., eventually clustering) the mean traversal length tends to be smaller for each ρ value. And, in Figure 2.21 (b) the mean traversal length increases as the ρ value increases.

In summary, we observe that from uniformness to clustering, traversal length tends to decrease. Hence, clustering obstacle patterns are not preferable in OPA's perspective. That is, OPA should avoid clustering obstacle patterns as much as possible and simply choose uniformness if uniformness and various clustering schemes



(a) (b)

Figure 2.20: (a) Mean traversal length versus ρ values for groups $r_0 = 5, 15, 30, 50$, (b) average contour plot of mean traversal length versus r_0 and ρ values.

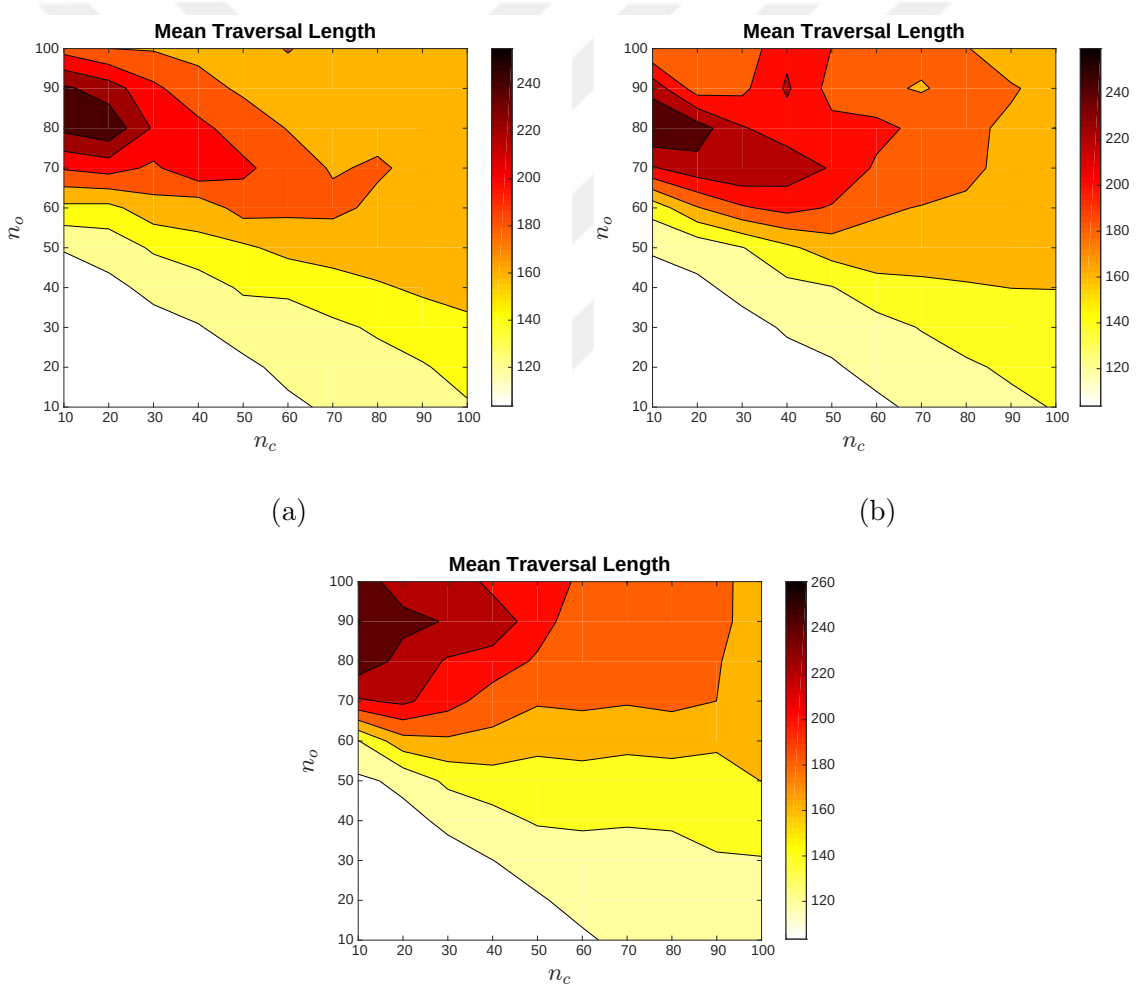


(a) (b)

Figure 2.21: (a) Mean traversal length versus ρ values for groups $\kappa = 1, 3, 6, 10$, (b) average contour plot of mean traversal length versus κ and ρ values.

are the only available options. So far, we have observed that the mean traversal length is maximized for moderate values of d (from 6 to 8) together with small values of γ (less than 0.1). We also observed that to maximize the traversal length of navigating

agent (NAVA), obstacle placing agent (OPA) should insert more true obstacles than false obstacles. Under these circumstances, we can also decide in what proportion of true obstacles to false obstacles should be used so as to maximize the traversal length of NAVA. We plot the mean traversal length versus n_o and n_c values for pairs $(d, \gamma) = (6, 0)$, $(d, \gamma) = (7, 0)$, and $(d, \gamma) = (8, 0)$. In Figure 2.22, we see that on the average the mean traversal length is maximized when n_o is around 80 together with n_c less than 20.



(a) (b)

(c)

Figure 2.22: Overall average contour plots of the mean traversal length versus n_o and n_c values when (a) $(d, \gamma) = (6, 0)$, (b) $(d, \gamma) = (7, 0)$, and (c) $(d, \gamma) = (8, 0)$.

2.7 Summary

We investigate how the traversal length of NAVA changes as the obstacle pattern changes from uniformness to regularity, and from uniformness to clustering for various pairs of numbers of clutter and true obstacles n_c, n_o. In our experimental setting, we simulated various combinations of both false and true obstacle number levels. Based on our investigation with extensive Monte Carlo simulations, we found that traversal lengths are higher under regularity than those under uniformness which tend to be higher that those under clustering.

Under regularity, we investigate the influence of the two parameters d and γ. Given the number of obstacles, the γ is the intensity of the number of pairs of distinct points lying closer than d units. In all cases, clutter only, true obstacle only and mixture of false and true obstacle, we recommend choosing moderate values of d (around 7) together with small values of γ (between 0 and 0.1).

Moreover, given the total number of obstacles, the ratio of true versus false obstacles is another important factor for the traversal length. Taking all of these into consideration, to achieve optimal value we recommend choosing moderate values of d (around 7) and smaller γ values (less than 0.1). And, given the total number of obstacles (i.e., $n = n_c + n_o$ is fixed), we also recommend choosing ρ value as large as possible in order to maximize the total traversal length of NAVA.

Under clustering obstacle pattern, the cluster radius r_0 and the number of parent points κ are found to be important. Here, the κ stands for the number of accumulation points and r_0 stands for the radius at which obstacles accumulate around the accumulation point. We do not recommend choosing clustering point pattern since it is not feasible. The number of true obstacles clearly dominates the number of false obstacles, again the parameter ρ plays an important role in determining the trend of traversal length when the total number of given obstacles is fixed. Hence, all things considered for optimal values we do not recommend using the clustering obstacle pattern in order to maximize the total traversal length of NAVA.

When the obstacle pattern and the number of obstacles kept same, then the traver-

sal length under cluttering pattern is stochastically less than or equal to the traversal length under uniformness, and which is also stochastically less than or equal to the traversal length under regularity. Thus, mean traversal length is maximized then the obstacle pattern is regular. Moreover, the traversal length in clutter only case is stochastically less than or equal to the traversal length in mixture case which is also stochastically less than or equal to the traversal length in true obstacle case only. Concerning the precision of sensor, as the sensor of NAVA increases the traversal length tends to be the more accurate and yields closer value to optimal solution.

In our future research, we are going to investigate the case where OPA places both true and false obstacles deterministically. We want to investigate whether knowing the exact locations of obstacles gains an advantage for OPA or not. Besides, concerning the theoretical results OPD has many similarities with classical CTP problem. We will prove our results for more general graphical settings by inspiring from current theoretical results belonging to CTP problem. In OPD problem only disambiguation capability was considered, but we also like to consider the problem with neutralization capability. Under feasible conditions, we will study the OPD problem by both disambiguation and neutralization capabilities together depending on the circumstances that NAVA and OPA encounters.

Another aspect of OPD problem is considering the study window in a 3-dimensional setting. That is to say, we would like to investigate the OPD problem in higher dimension as well starting from three dimensional case. In 3-D case, NAVA may represent the submarine that wishes to reach from one source point to a target destination avoiding underwater mines. And, OPA may be the opponent forces wishing to place mines so as to maximize the total traversal length of NAVA.

Chapter 3

M2^K ALGORITHM FOR THE OPD PROBLEM

3.1 Introduction

In this section, we investigate how traversal length behaves for various M2^k algorithms. We will use the same experimental setup of Aksakalli and Ceyhan (2012), but compute the traversal length by M2^k algorithm for various integer values of k. Based on our Monte Carlo simulations, we observe that the trends for mean traversal length computed by RD algorithm and M2^k algorithm are essentially similar. So, instead using the greedy algorithm we recommend using M2^k algorithm where few disambiguations take place, and hence decrease the complexity cost compared to the original RD algorithm.

3.2 Experimental Setting

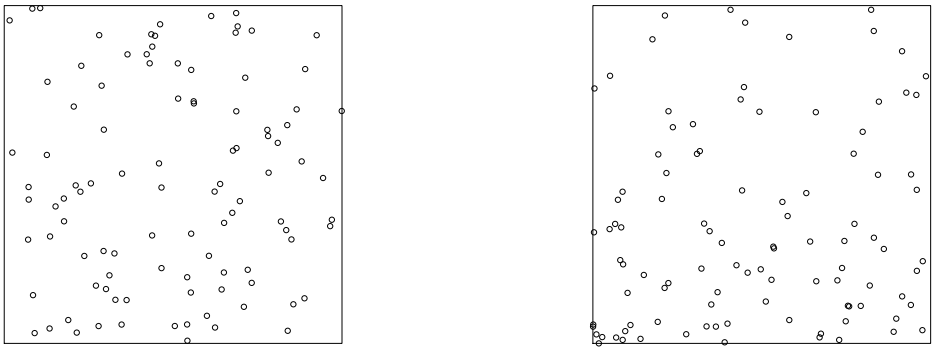
3.2.1 Background Clutter Generation

In our simulation setting, we use 6 different types of point processes for sampling centers of disk-shaped clutters. These are homogeneous (HPP) and inhomogeneous (IP) Poisson processes, Matérn and Thomas point processes, and Hardcore and Strauss point processes. Among these Matérn and Thomas processes are clustered point processes, while Hardcore and Strauss processes are regular point processes (Baddeley, 2010). The number of clutter is fixed to 100, and the number of obstacles are 20,30,40,50 and 60 respectively. As for obstacle window forms, we use a total of 19 different obstacle placement patterns. Obstacles are uniformly sampled within four different window forms: the entire background window itself (P), linear strips (L), V-shaped (V) and W-shaped (W) polygonal windows.

Formally, a *spatial point process* X is a finite random subset of a bounded region $\Omega \subset \mathbb{R}^2$. A realization of this point process, on the other hand, is called a *spatial point pattern*. Mainly there are three types of spatial point patterns; independent patterns, cluster patterns where points tend to be close to one another and regular patterns where points tend to repel each other. In this section, we consider two patterns from each one of these three groups and the background clutter disk centers are generated from those six spatial point processes (Aksakalli and Ceyhan (2012)).

The homogeneous Poisson process with constant intensity λ is also called complete spatial randomness (CSR), where the intensity will be denoted by CSR(λ). For any region Ω, the CSR point process has four properties: (1) the number of points in Ω is a Poisson random variable, (2) number of points in any two disjoint regions Ω and Ω' are independent random variables, (3) the expected number of points in Ω is $\lambda \times area(\Omega)$, and (4) points in Ω are independently and uniformly distributed (given the number of points). In our case, λ is taken to be 100. As for the inhomogeneous Poisson process, it is a modification of CSR where the intensity is not constant, but varies from location to location. Specifically, the intensity is a function in two dimensional Euclidean space. Let $IP(\lambda(h))$ denote the inhomogeneous Poisson process with intensity $\lambda(h)$ where $h \in \mathbb{R}^2$. Here, the intensity function $\lambda(h)$ specifies the value of λ on the plane. Properties of $IP(\lambda(h))$ are the same as those of CSR(λ) with the last two properties modified as follows: (3′) the expected number of points in Ω is $\int_\Omega \lambda(h)dh$, and (4′) points in Ω are independently identically distributed (i.i.d) with probability density $f(h) = \lambda(h) \left[\int_\Omega \lambda(h)dh\right]^{-1}$. In our case, we take the intensity function as $\lambda(x, y) = 0.037e^{(10-y)/40}$ as in Aksakalli and Ceyhan (2012). Notice that, the intensity only depends on y and decreases as y increases. Here, obstacles get denser as one gets closer to a target point. An illustration of both CSR and IP is shown in Figure 3.1.

For clustering, the Matérn point process, denoted by $M(\lambda, \mu, \sigma)$, is constructed by first generating a Poisson point process of "parent" points with intensity λ. Each parent point is then replaced by a random cluster of points where the number of points in each cluster is sampled from a Poisson distribution with parameter μ. These so

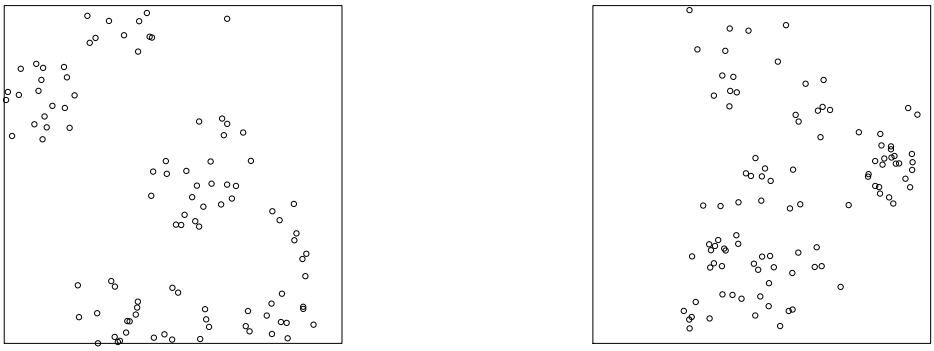


(a) CSR (b) Inhomogeneous Poisson

Figure 3.1: Sample realizations from homogeneous and inhomogeneous point processes. The parameters are: (a) $CSR(100)$ and (b) $IP\left(0.037e^{(10-y)/40}\right)$. Indeed, on the average 100 points are generated by using these distribution parameters, but by using the rejection sampling we fix them to exactly 100 points.

called "child" points are placed independently and uniformly inside a disk with fixed radius r centered at the parent point. In our case, we work with $M(10, 10, 10)$. Similar to the Matérn point process, the Thomas process, denoted by $T(\lambda, \mu, \sigma)$, is constructed by first generating a Poisson point process of "parent" points with intensity λ. A random cluster of points replaces each parent point with the number of points per cluster being sampled from a Poisson distribution with parameter μ. In contrast with the Matérn point process, positions of these child points in the Thomas point process are isotropic Gaussian displacements centered at the cluster parent location with standard deviation σ. In our case, we work with $T(10, 10, 5)$ as in Aksakalli and Ceyhan (2012). An illustration of both Matérn and Thomas is shown in Figure 3.2.

For regular, the probability density function of the hardcore process is that of the Poisson process with intensity function λ conditioned on the event that no two points generated by the process are closer than d units apart, hence denoted by

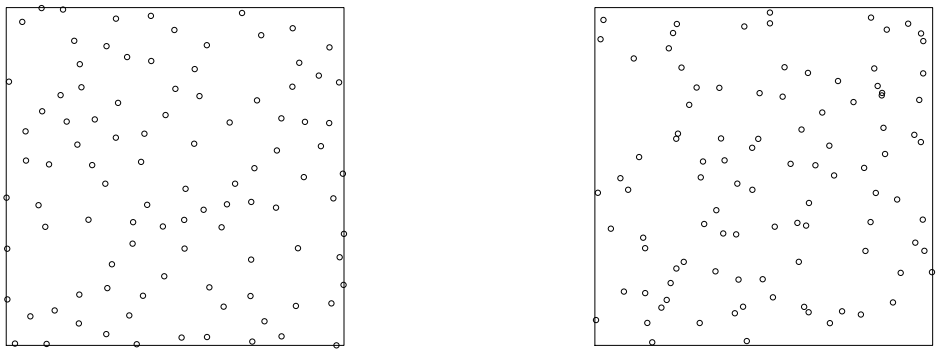


(a) Matérn (b) Thomas

Figure 3.2: Sample realizations from Matérn and Thomas point processes. The parameters are: (a) $M(10, 10, 10)$ and (b) $T(10, 10, 5)$. Indeed, on the average 100 points are generated by using these distribution parameters, but by using the rejection sampling we fix them to exactly 100 points.

$HC(\lambda, d)$. The Strauss process, denoted $S(\lambda, d, \gamma)$, on the other hand, generalizes the hardcore process by incorporating a $\gamma \in [0, 1]$ parameter that controls of the interaction between points. The process exhibits more regularity for smaller values of γ, and less regularity for large γ. For $\gamma = 0$, the Strauss process becomes hardcore process, and for $\gamma = 1$, it reduces to CSR. In our case, we work with $HC(100, 5)$ and $S(100, 5, 0.5)$ as in Aksakalli and Ceyhan (2012). An illustration of both Hardcore and Strauss is shown in Figure 3.3.

In our computational experiment (see Section 2.2), the graph $G = (V, E)$ is the 8-adjacency integer discretization of $[0, 100] \times [0, 100]$ with $s = (50, 100)$ and $t = (50, 1)$. Each disk has a fixed radius of $r = 4.5$ units and the disk centers are sampled inside $[10, 90] \times [10, 90]$ ensuring that there always exists(possibly very long) a s, t walk. The disambiguation cost is taken to be $c = 5$. And, clutter probabilities are sampled from Beta(2,6), whereas obstacle probabilities are sampled from Beta(6,2). These choices



(a) Hardcore (b) Strauss

Figure 3.3: Sample realizations from Hardcore and Strauss point processes. The parameters are: (a) $HC(100, 5)$ and (b) $S(100, 5, 0.5)$. Indeed, on the average 100 points are generated by using these distribution parameters, but by using the rejection sampling we fix them to exactly 100 points.

are made in line with the ones in Aksakalli and Ceyhan (2012) and Priebe et al. (2005). Also, we let the significance level of the test be 0.01.

In particular, let $P = [10, 90] \times [10, 90]$ which will be the window for clutter center points. In all six types of clutter distributions, the sample size generated is equal to exactly 100 points. As for obstacle window forms, we consider a total of 19 different sampling windows. The first is the polygon $P = [10, 90] \times [10, 90]$, there are 8 different linear windows of width equals to 10 units and with their top left corner y coordinate being 90,80,...,20, and 5 different V-shaped and W-shaped windows, respectively, with their top left corner y coordinate being 90,80,...,50. The difference between the top and bottom y coordinates of each one of these 10 polygons is taken to be 50 units. The width of polygonal shape is 10 units (Aksakalli and Ceyhan, 2012).

For example, L70 is the polygon whose four corner points are (10,70), (90,70), (90,60) and (10,60) clock-wise starting with the top left corner. We also take number

of obstacles to be 20, 30, 40, 50, and 60 for each of the 19 obstacle sampling windows. Thus, the realization of obstacle patterns are denoted by the sampling window followed by the number of obstacles. For example, P:40 and L70:40 refer to the obstacle patterns sampled within the P and L70 windows, respectively, with 40 obstacle center points inside their respective windows. These choices also ones made as in Aksakalli and Ceyhan (2012) for comparative purposes. Hence, the treatment factors are the "background clutter type", the "obstacle placement window" and the "number of the obstacles". The first has 6 different types, the second has 19 types and the third has 5 different types resulting in a total of 570 treatment combinations. Our main goal is to investigate how traversal length changes according to treatment factors when the NAVA uses the greedy RD algorithm and its variants M2^k algorithm.

3.3 M2^k Algorithm

In the original version of RD algorithm, the average number of edges that disk-shaped obstacles intersect in a 8-discretization setting is 88. So, prior to the traversal, NAVA assigns edge weight to all edges intersecting the disks using the cost of disambiguation and the probability of being a true obstacle of a disk (Chapter 5). But, M2^k algorithm is based on the effective choice of the number disambiguations. As an example let us consider the case $k = 3$, then NAVA updates only 2^3 edges for each disk-shaped obstacle and assigns ∞ for the rest. An example of the choice of 8 disambiguation points for disks are shown below (see Figure 3.3). M16 and M4 algorithms are defined analogously. The advantage of using the M2^k algorithm is that the complexity time is reduced when computing the traversal length of NAVA and the mean traversal length is well estimated. For example, the complexity time of running M16 algorithm is at least 5 times faster than the RD algorithm, and the mean traversal length estimated by M16 algorithm deviates at most 2.5% relative error from the traversal length estimated by RD algorithm.

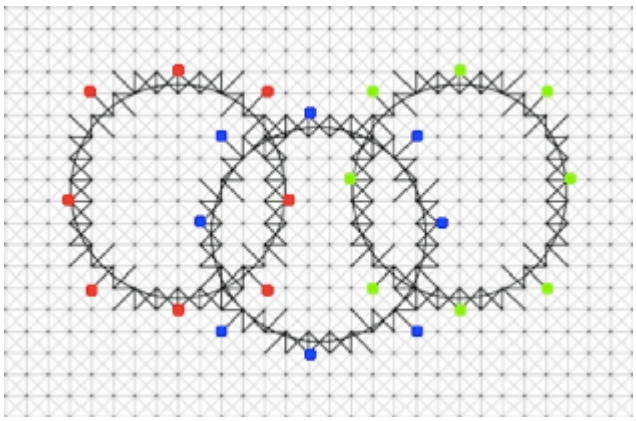


Figure 3.4: A total of 8 disambiguation points are labeled in each disk-shaped obstacle.

3.4 Repeated Measures ANOVA

Our data set consists of traversal lengths computed by the RD algorithm and its variants under various conditions that we have described above. Hence, the background clutter types are Complete Spatial Randomness (CSR), inhomogeneous Poisson (IP) pattern, Matérn (M) pattern, Thomas (T) pattern, hardcore (HC) pattern, and Strauss (S) pattern. For simplicity the clutter types are numbered from 1 to 6, which in turn, correspond to CSR, IP, M, T, HC, and S patterns, respectively. Similarly, the obstacle window types are numbered from 1 to 19 corresponding to P, L90, L80,..., L20, V90, V80,..., V50, W90, W80,..., W50, respectively. The obstacle number levels are also numbered from 1 to 5 corresponding to 20, 30, 40, 50, 60, respectively. Thus, we denote the traversal length as L_{ijkl} which is the traversal length of the measurement l for clutter type i, obstacle window type j, obstacle number level k with $l = 1, 2, ..., 100, i = 1, 2, ..., 6, j = 1, 2, ..., 19$, and $k = 1, 2, ..., 5$, respectively.

Note that $L_{ijkl}, L_{ij'kl}, L_{ijk'l}, L_{ij'k'l}$ are estimated on the same background clutter type i, and hence they are potentially correlated. Similarly, the expected traversal lengths within the respective obstacle window types or obstacle number levels can also correlated (positively or negatively). In order to deal with and take into account such possible correlations, we use repeated measures ANOVA. We use the repeated mea-

sures ANOVA to compare the traversal length differences between treatment factors. The conditions of repeated measures ANOVA are similar to that of usual ANOVA, except that independence is not required and an assumption about the relations among the repeated measures is added. The repeated measures ANOVA assumptions are; (i) the dependent variable is normally distributed, (ii) the homogeneity of covariance matrices is required, (iii) independence between predictor factors, and (iv) sphericity, which refers that the variance of repeated measures are all equal, and the correlations among the repeated measures are all equal (Aksakalli and Ceyhan, 2012). The main advantage of using the repeated measures ANOVA rather than standard ANOVA is that we obtain more precise information in comparison of our results. Moreover, as mentioned in Aksakalli and Ceyhan (2012), we consider four types of variance-covariance (var-cov) structure; (i) unstructured (UN), (ii) autoregressive (AR1), (iii) autoregressive heterogenous (ARH1), and (iv) compund symmetry (CS).

First we investigate the type(s) and level(s) of interaction for each pair of treatment combination factors. Consider the following model with four different variance-covariance (var-cov) structures:

$$L_{ijkl} = \mu_0 + \mu_i^C + \mu_j^O + \mu_l^{NO} + \epsilon_{ijkl} \tag{3.1}$$

where μ_0 is the overall mean, μ_i^C is the mean for clutter type i, μ_j^O is the mean for obstacle type j, μ_i^{NO} is the mean for obstacle number level k, and ϵ_{ijkl} is the error with $l = 1, 2, ..., 100, i = 1, 2, ..., 6, j = 1, 2, ..., 19$, and $k = 1, 2, ..., 5$, respectively. The var-cov structures we consider are compound symmetry (CS), autoregressive var-cov structure (AR1), autoregressive var-cov structure heterogeneous (i.e., with different variances) at obstacle types (ARH1-OT), and autoregressive var-cov structure heterogeneous (i.e., with different variances) at obstacle number levels (ARH1-ON).

3.4.1 Overall Comparison of Traversal Lengths

Analogous to Aksakalli and Ceyhan (2012), let us investigate the type and levels of interaction for each pair of treatment combination factors by using the model given in Equation 3.1. The model comparisons (with degrees of freedom (df), AIC, -log

likelihood values) are presented in Table 3.1 for the traversal length estimated by RD algorithm. The model comparisons for traversal length estimated by M16, M8 an M4 are also similar, and hence we do not provide them separately. Note that the model with ARH1-OT var-cov structure seems to be the best model, since it has the smallest AIC smallest -log likelihood values. Similarly, ARH1-OT values when the traversal length is estimated by M16, M8 and M4 algorithms are 549741.70, 528238.90 and 496291.10, respectively. The p-values shown in Table 3.1 are based on the likelihood ratio of the best model (which is ARH1-OT) and the model in the corresponding row. With ARH1-OT model, we observe significant differences between clutter types, obstacle type, and obstacle number levels. The interaction plots RD and M8 algorithms are shown in Figures 3.5, 3.6. The interaction plots of M16 and M4 are similar, and hence omitted.

var-cov structure	df	AIC	-log likelihood	L-ratio	p-value
CS	30	587112.9	293526.5	11760.17	<0.0001
AR1	30	587102.6	293521.3	11749.83	<0.0001
ARH1-ON	34	581590.7	290761.4	6229.987	<0.0001
ARH1-OT	48	575388.7	287646.4	-	-

Table 3.1: The comparisons of the models in Equation (1) when the traversal length is estimated by RD algorithm.

RD algorithm case:

When obstacle number levels are ignored (i.e., when only interaction between obstacle type/forms and clutter types are considered), we find that obstacle and clutter types do not have significant interaction ($p = 0.0214$), neither do obstacle forms and clutter types ($p = 0.122$) which means that the trends in mean length plotted in Figures 3.5(a)-(b) and are not significantly different from being parallel. Hence, it is reasonable to compare the main effects of clutter and obstacle types on the traversal length. According to our simulation results, we find that the traversal lengths are

significantly different between background clutter types ($p < 0.0001$) and between obstacle types ($p < 0.0001$). Similarly, traversal lengths are significantly different between clutter types ($p < 0.0001$) and between obstacle forms ($p < 0.0001$). Also note that, on the average at each obstacle type or form, Matérn and Thomas clutter types tend to yield shorter traversal lengths; whereas Hardcore and Strauss clutter types tend to yield longer traversal lengths. Hardcore clutter tends to yield the longest traversal lengths, which suggests that the more regular the clutter type, the longer the traversal lengths. Moreover, at each obstacle type or form, the average traversal lengths (in ascending order) are for P, W-shaped, linear, and V-shaped obstacle forms. When clutter types are ignored (i.e., when only interaction between obstacle types/forms and obstacle number levels are considered), we find significant interaction between obstacle type and obstacle number levels ($p < 0.0001$), and between obstacle form and obstacle number levels ($p < 0.0001$), which means the trends in mean traversal lengths plotted in Figures 3.5(d)-(e) are significantly non-parallel. Hence, it is not reasonable to compare the mean traversal lengths for obstacle types/forms and obstacle number levels, but instead, for example, it will make sense to compare the mean traversal lengths for obstacle number levels at each obstacle type or form. At P and W-shaped obstacle forms, traversal lengths tend to increase as the obstacle number increases; at linear and V-shaped obstacle forms, traversal lengths exhibit a concave down trend (i.e., increase, reach a peak, and then decrease); for linear and V-shaped windows the shortest lengths occur at 20 obstacles, but longest lengths occur at 40 and 50 obstacles, respectively. This behavior is due to the increase in the disk-shaped obstacles (true obstacle and clutter) density that makes the NAVA to decide to traverse along the boundary more often. Moreover, for 20 and 30 obstacles, the largest (on the average) traversal lengths occur for linear obstacle forms, and for 40-60 obstacles, longest occurs for V-shaped obstacle forms. At each obstacle number level, the shortest traversal lengths occur for the P obstacle form.

When obstacle types are ignored (i.e., when only interaction between clutter type and obstacle number levels are considered), we find significant interaction between

clutter type and obstacle number levels ($p < 0.0001$), which means the trend in mean traversal length plotted in Figure 3.5(c) is significantly non-parallel. Hence we do not test for the main effects of clutter types and obstacle number levels. Instead, we compare the mean traversal length values for obstacle number levels at each clutter type. On the average at each clutter type, traversal lengths tend to increase as the obstacle number increases (up to 40 obstacles) but the lengths for 50 and 60 obstacles are very similar. At each obstacle number level, the longest traversal lengths occur for Hardcore clutter type, and the shortest traversal lengths occur for Thomas clutter types.

The shortest and longest traversal length performances are presented in Table 3.2. In our overall comparison, the shortest length is about 118 units which occurs at T:P:20, T:W90:20, M:W80:20, and M:P:20 treatment combinations, and the longest length is about 206 units which occurs at HC:V80:50 treatment combination.

M16 algorithm case:

When the other factor is ignored, we find significant interactions for all pair of treatment factors plotted as in Figures 3.5, 3.6. In all those subfigures, their respective $p-$values are strictly less than 0.0001, which means the trend in mean traversal length are significantly non-parallel. On the average, Hardcore clutter types tend to yield longer traversal lengths and Matérn or Thomas clutter types tend to yield shorter traversal lengths. Also, $V-$shaped obstacle forms tend to dominate others for each clutter type and obstacle number levels except 20 and 30 obstacle number levels. At P and W-shaped obstacle forms, traversal lengths tend to increase as the obstacle number increases; at linear and V-shaped obstacle forms, traversal lengths exhibit a concave down trend (i.e., increase, reach a peak, and then decrease).

We record the shortest and longest traversal length performances as in Table 3.2. In our overall comparison, the shortest length is about 119 units which occurs at M:W:80:20 and T:W90:20 treatment combinations, and the longest length is about 186 units which occur at HC:V:50:50 and T:W90:60 treatment combinations.

M8 algorithm case:

When obstacle number levels are ignored (i.e., when only interaction between obstacle type/forms and clutter types are considered), we find significant interaction between obstacle type and clutter type ($p = 0.0024$), which means the trend in mean traversal length plotted in Figure 3.6(a) is significantly non-parallel. Hence, it is not reasonable to compare the mean traversal lengths for obstacle types and clutter types, but instead, for example, it will make sense to compare the mean traversal lengths for obstacle types at each clutter type. Similarly, when the other factor is ignored, we find significant interactions for all pair of treatment factors plotted in Figure 3.6. In all four sub-cases, their respective $p-$values are strictly less than 0.0001. As it was in RD algorithm, on the average Hardcore clutter types tend to yield longer traversal lengths and Matérn and Thomas clutter types tend to yield shorter traversal lengths. Also, unsurprisingly $V-$shaped obstacle forms tend to dominate others for each clutter type and obstacle number levels. Although Hardcore clutter type and $V-$shaped forms tend to give longer traversal lengths on the average as the obstacle number levels increase, it is crucial to note that after certain point all of them give close values.

The shortest and longest traversal length performances are presented in Table3.3. In our overall comparison, the shortest length is about 121 units which occurs at M:P:20 and M:W:80:20 treatment combinations, and the longest length is about 177 units which occur at IP:V:50:60 and HC:V70:40 treatment combinations.

M4 algorithm case:

When the other factor is ignored, we find significant interactions for all pair of treatment factors plotted as in Figures 3.5, 3.6. In all those subfigures, their respective $p-$values are strictly less than 0.0001, which means the trend in mean traversal length are significantly non-parallel. On the average Hardcore clutter types tend to yield longer traversal lengths and Matérn or Thomas clutter types tend to yield shorter traversal lengths. Also, $V-$shaped obstacle forms tend to dominate others for each clutter type and obstacle number levels except for CSR clutter type and 20 obstacle number level. At each clutter type and obstacle form, on the average the traversal lengths increase as the number of obstacles increase. Moreover, when obstacle number

level ignored it can be seen from the graph that the trend in traversal lengths is constant for all regular background clutter type at each obstacle form.

We record the shortest and longest traversal length performances as in Table 3.2. In our overall comparison, the shortest length is about 122 units which occurs at M:W:80:20 and M:P:20 treatment combinations, and the longest length is about 174 units which occur at IP:V:50:60 and HC:V50:60 treatment combinations.

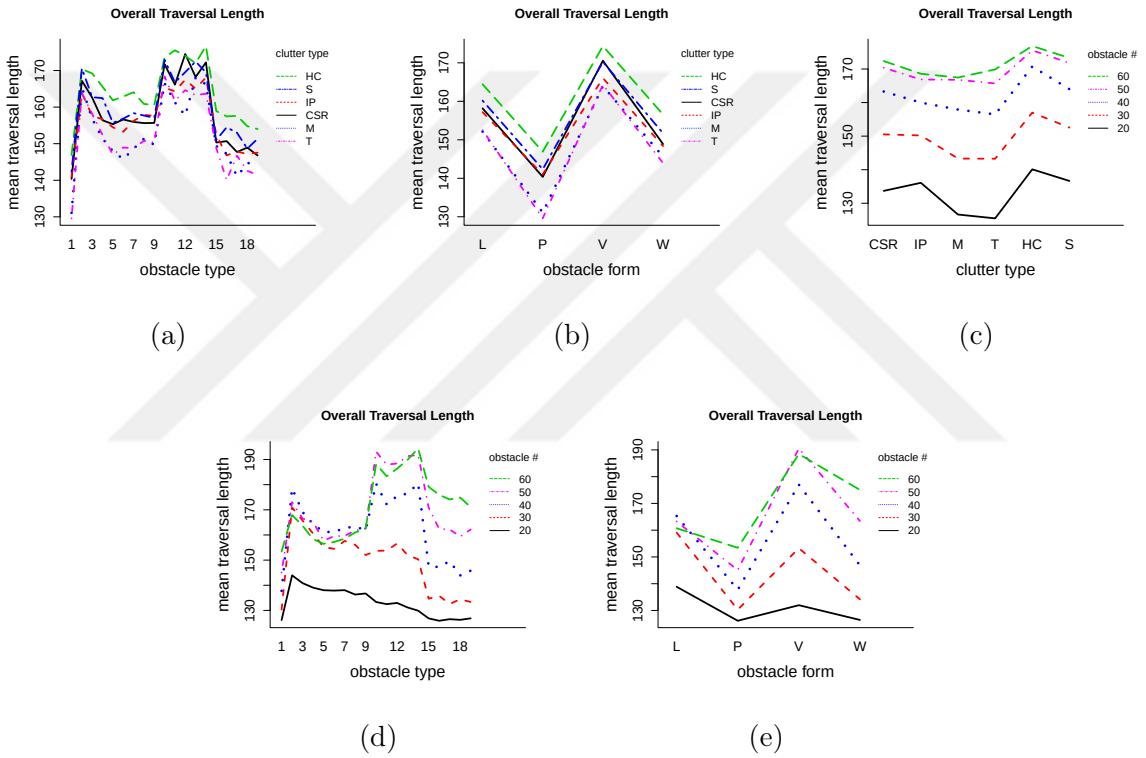


Figure 3.5: The interaction plots for each pair of treatment factors (obstacle type/form, clutter type, and obstacle number) when the other factor is ignored. The traversal length is estimated by RD algorithm.

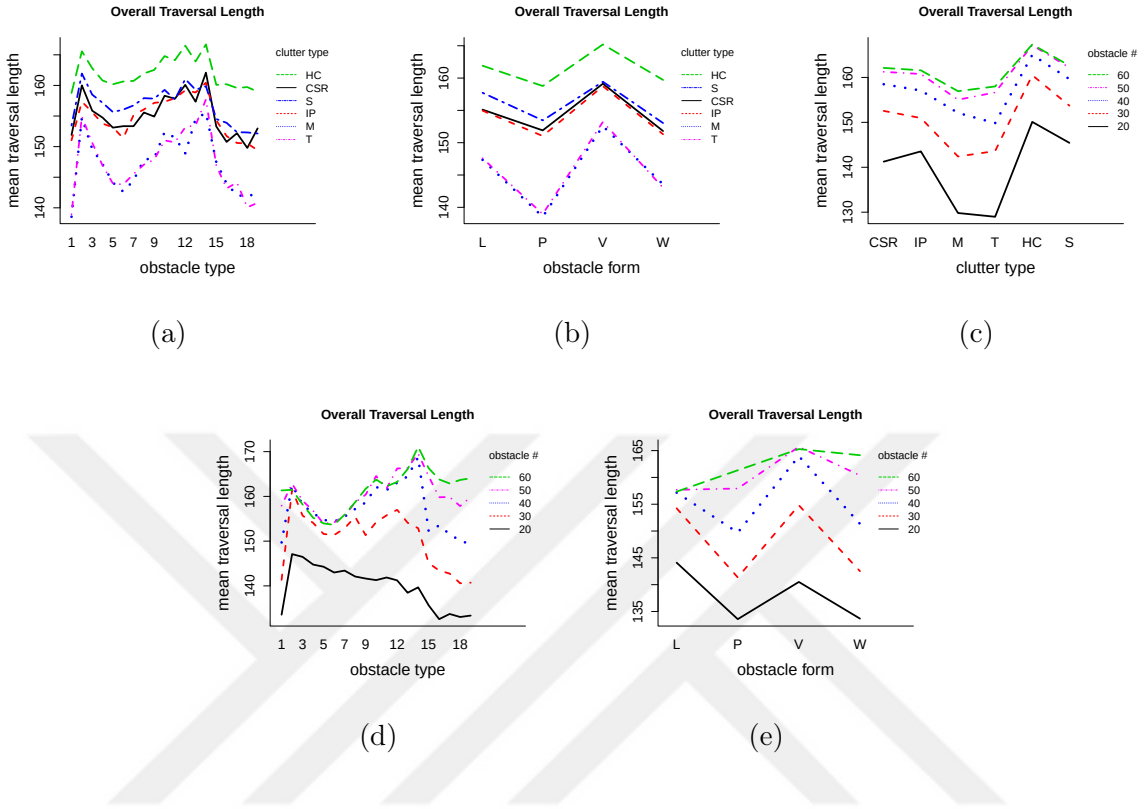


Figure 3.6: The interaction plots for each pair of treatment factors (obstacle type/form, clutter type, and obstacle number) when the other factor is ignored. The traversal length is estimated by M8 algorithm.

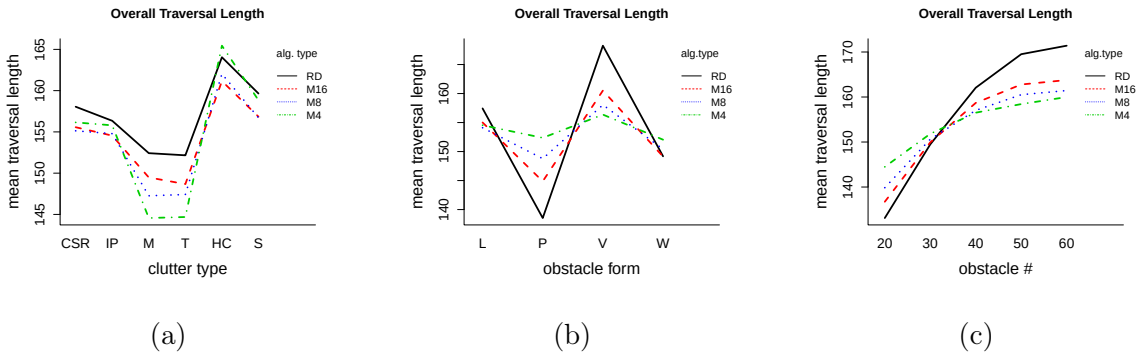


Figure 3.7: The plots of mean traversal length versus obstacle window type, clutter type, and obstacle number level when computed by RD, M16, M8, and M4 algorithms, respectively.

Overall-RD algorithm				
	SHOSTEST		LONGEST	
	traversal length(s)	treatment types(s)	traversal length(s)	treatment type(s)
RD Algorithm	117.87, 118.03 118.08, 118.10	T:P:20, T:W90:20 M:W80:20, M:P:20	206.73	HC:V80:50
Clutter type				
CSR	125.06	W80:20	200.65, 200.60	V50:60, V70:60
Inhom. Poisson	126.93	W60:20	196.03, 187.44	V50:60, V60:60
Matérn	118.08, 118.10	W80:20, P:20	196.21, 195	V60:60, V60:50
Thomas	117.87, 118.03	P:20, W90:20	199.06	V90:60
Hardcore	134.71 134.84, 135.39	W90:20, V50:20 W70:20, P:20	206.73	V80:50
Strauss	128.84, 129.68	W80:20, P20	195.82, 195.15	V90:50, V90:60
Obstacle form				
P	117.87, 118.1	T:20, M:20	160.52	HC:60
Linear	127.13, 128.46	L30:T:20, L40:M:20	186.05, 185.23	L90:S:40, L90:HC:40
V-shaped	120.65	V50:T:20	206.73	V80:HC:50
W-shaped	118.03, 118.08	W90:T:20, W80:M:20	191.73, 186.02 184.21, 184.15	W90:T:60, W80:HC:60 W90:HC:60, W80:CSR:60
Number of Obstacles				
20	117.87, 118.03 118.08, 118.10	T:P:20, T:W90:20 M:W80:20, M:P:20	154.21, 148.79 148.79, 146.32 146.15	IP:L90:20, IP:L80:20 IP:L80:20, HC:L80:20 S:L90:20
30	121.13, 121.49	M:P:30, T:P:30	177.34	HC:L80:30
40	127.65, 129.03	T:P:40, M:P:40	192.58	HC:V70:40
50	136.73	M:P:50	206.73	HC:V80:50
60	143.74	T:P:60	200.65, 200.64	CSR:V50:60, HC:V50:60

Table 3.2: The shortest and longest traversal lengths and the corresponding treatment types for overall comparisons, and comparisons at specific clutter types, obstacle types, and obstacle numbers when the traversal length is computed by RD algorithm.

Overall-M8 algorithm				
	SHOSTEST		LONGEST	
	traversal length(s)	treatment types(s)	traversal length(s)	treatment type(s)
M8 Algorithm	121.58, 121.82 122.56, 122.74	M:P:20, M:W80:20 T:W60:20, T:P:20	177.11, 175.79	IP:V50:60, HC:V70:40
Clutter type				
CSR	131.29	W80:20	171.71, 170.59	V50:40, V70:50
Inhom. Poisson	133.63	W50:20	177.11, 173.91	V50:60, V50:50
Matérn	121.58, 121.82	P:20, W80:20	168.59, 167.89	V50:60, V50:40
Thomas	122.56, 122.74	W60:20, P:20	175.01, 171.04	W90:60, V50:40
Hardcore	144.45, 144.97	P:20, W80:20	175.79, 175.04	V70:40, V50:50
Strauss	136.45, 137.35	W80:20, P20	169.15, 167.98	V50:60, V70:40
Obstacle form				
P	121.58, 122.74	M:20, T:20	169.57, 165.44	HC:60, HC:50
Linear	140.14	L50:M:30	166.71, 165.45	L90:HC:30, L90:CSR:30
V-shaped	123.84	V60:T:20	177.11, 175.79	V50:IP:60, V70:HC:40
W-shaped	121.82, 122.56	W80:M:20, W60:T:20	175.01, 172.67	W90:T:60, W90:HC:60
Number of Obstacles				
20	121.58, 121.82	M:P:20, M:W80:20	155.63, 154.70	HC:L80:20, HC:L50:20
30	127.60, 128.81	M:P:30, T:W50:30	167.29	HC:V80:30, HC:V90:60
40	133.84	T:P:40	175.79	HC:V70:40
50	145	T:V90:30, T:V60:50	175.04, 174.30	HC:V50:50, HC:V80:50
60	145.61	T:L50:60	177.11, 175.01	IP:V50:60, T:W90:60

Table 3.3: The shortest and longest traversal lengths and the corresponding treatment types for overall comparisons, and comparisons at specific clutter types, obstacle types, and obstacle numbers when the traversal length is computed by M8 algorithm.

3.5 Summary

In this work we investigated the differences in trends of traversal length of NAVA computed by M16, M8, and M4 algorithms compared to the traversal length computed by RD algorithm carried out by Aksakalli and Ceyhan (2012). To gain more power and precision in our analysis, we used similar statistical analysis tools, i.e., the repeated measures of ANOVA. Extensive Monte Carlo simulations indicate that as the background clutter distribution gets more regular (clustered), on the average, the traversal length tends to be longer (shorter) when M2^k algorithm is deployed for values of $k = 2, 3, 4$. In terms of shortest and longest traversal lengths, the trends are essentially same for each M16, M8, M4 algorithms compared to RD algorithm i.e., the shortest occur when the background clutter is clustered, and the longest traversal length occur when the background clutter type is from regular such Hardcore point process. Moreover, the main advantage of using M2^k algorithms, for example M16 algorithm, that the complexity time is at least 5 times faster than RD algorithm and the computed traversal length is within 2.5% error.

Moreover, the similar trends among RD and M2^k can also be observed in Figure 3.7 at each background clutter type, obstacle window form, and obstacle number levels. For each algorithm, the mean traversal length tends to be higher when the background clutter type is regular (Figure 3.7(a)), or when true obstacles are uniformly distributed inside V-shaped obstacle window form (Figure 3.7(b)), and the mean traversal length tends to increase as the number of obstacles increase (Figure 3.7(c)).

Chapter 4

DEPENDENCE OF TRAVERSAL LENGTH ON OBSTACLE WINDOW TYPES AND TESSELLATIONS

4.1 Introduction

In this Chapter, we investigate how traversal length depends on obstacle window forms. In Aksakalli and Ceyhan (2012) a few obstacle forms were considered, but in this article we address more types of obstacle forms. For linear strips, we change the width and the location of strip (from target to source point). For V-shaped obstacle forms, we change the distance between the upper lips of V-shape and the location of V-shaped obstacle form. And for semi-circle obstacle form the inner and the outer radii are fixed, we only change the location of semi-circle obstacle form.

In Chapter 2, we have observed that when the entire working space is equipped with only false obstacles, then the Hardcore point pattern (regular) tends to yield larger traversal length. Thus, for a given fixed number of false obstacles we use the Hardcore point pattern for clutter points. And, true obstacles inside the obstacle forms will be distributed uniformly. Moreover, we will investigate the case where true obstacles are distributed inside obstacle window forms as regular as possible using the Strauss point pattern.

Moreover, we investigate the case where OPA knows the exact locations of obstacles. We study the trends in mean traversal length when the true obstacles are placed randomly proportional to the areas of Voronoi polygons or Delaunay triangles based on the allocation of the clutter points. We investigate the case when the true

obstacles are placed around the vertices of Voronoi polygons or around the centroids of Delaunay triangles. By performing these operations we aim to cover the study window as far as possible, and thus maximize the traversal length of NAVA. Based on our Monte Carlo simulations, on the average the mean traversal length is essentially the same as the mean traversal length when obstacles are distributed completely random (uniformness).

In our computational experiment (see Section 2.2), the graph $G = (V, E)$ is the 8-adjacency integer discretization of $[0, 100] \times [0, 100]$ with $s = (50, 100)$ and $t = (50, 1)$. Each disk has a fixed radius of $r = 4.5$ units and the disk centers are sampled inside $[10, 90] \times [10, 90]$ ensuring that there always exists(possibly very long) a s, t walk. The disambiguation cost is taken to be $c = 5$. And, clutter probabilities are sampled from Beta(2,6), whereas obstacle probabilities are sampled from Beta(6,2). These choices are made in line with the ones in Aksakalli and Ceyhan (2012) and Priebe et al. (2005).

4.2 Obstacle Window Types

4.2.1 Linear Strip Window

In the OPD problem, we investigate how the traversal length varies as the obstacle window form changes. Let denote by n_c and n_o the number of false and true obstacles, respectively. In Aksakalli and Ceyhan (2012) n_c was equal to 100 and n_o values were $20, 30, 40, 50, 60$. We will use the same numbers, but in order to maximize the mean traversal length we will use Hardcore(n_c, d) (Strauss(n_c, d, γ) point process with $\gamma = 0$) point process for clutter pattern with parameters $n_c = 100$ and $d = 5, 6, 7, 8$ (Chapter 2). As for true obstacles, the centers of disk-shaped obstacles will be distributed uniformly inside the linear strip with width ℓ (Figure 4.1). By using the same Hardcore point process for true obstacles, we will also investigate the case when true obstacles are distributed as regular as possible inside linear strips. Together with the width ℓ of linear strip, we also change the location of strip; close to target, at center, and close to source. Let L10, L50, L90 be linear strips locating at the bottom, center, and at

the top in an working environment. Here, L10 refers to the leftmost bottom corner of strip that is close to target, L50 refers to the midpoint of left width of strip locating at center, and L90 refers to the leftmost upper corner of strip that is close to source.

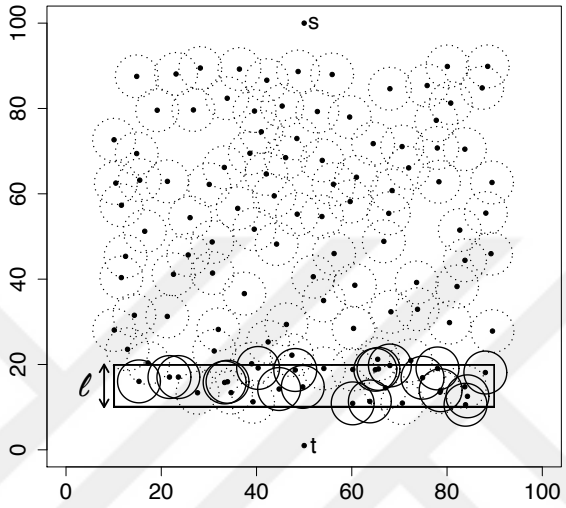
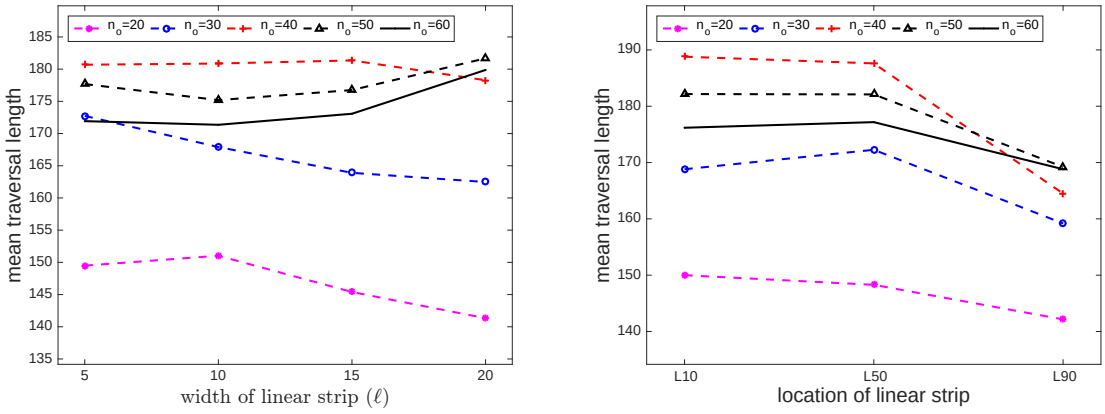


Figure 4.1: The background clutter pattern is from the Hardcore$(100, 5)$ (dashed circles) process and $n_o = 20$ true obstacles are uniformly distributed inside the linear strip of width $\ell = 10$ (solid circles).

In the work of Aksakalli and Ceyhan (2012), the longest traversal length occurred when the background clutter type was of Hardcore$(100, 5)$ point process with $n_o = 40$ true obstacles uniformly distributed inside the linear strip of width $\ell = 10$ locating at the bottom of working space (i.e., close to target).

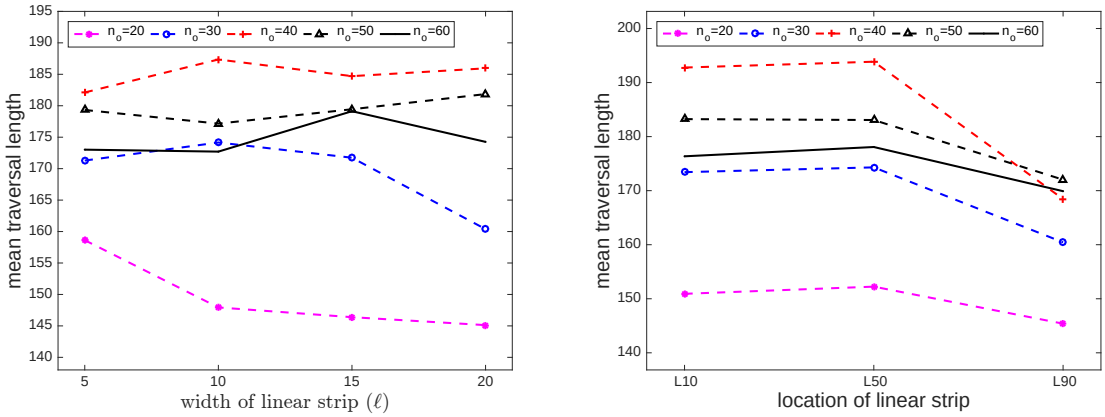
In Figure 4.2 we observe that the traversal length is maximized when the obstacle number level is $n_o = 40$ provided that the width of linear strip ℓ is less than or equal to 15 and that linear strip locates at the bottom in an environment. In Figure 4.2 the background clutter pattern is from Hardcore$(100, 5)$, but we will also investigate the trend of traversal length when the background clutter changes as well.

Next, In Figure 4.3 the traversal length is again maximized when the obstacle number level is $n_o = 40$ regardless of width of linear strip ℓ, but this time the traversal length is larger than the case when the clutter pattern is Hardcore$(100, 5)$. When



(a) (b)

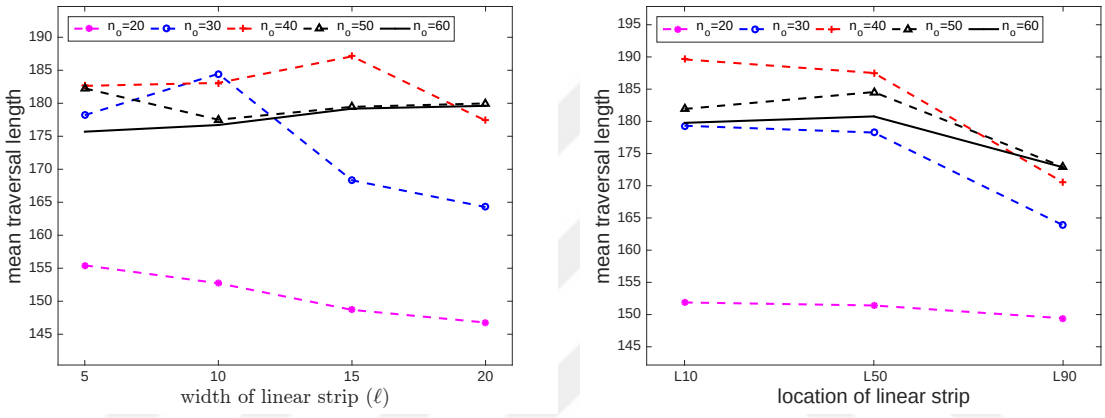
Figure 4.2: Mean traversal length versus (a) width ℓ of linear strip and (b) location of linear strip for $n_o = 20, 30, 40, 50, 60$ provided that background clutter type is Hardcore$(100, 5)$. Notice that vertical axes are differently scaled.



(a) (b)

Figure 4.3: Mean traversal length versus (a) width ℓ of linear strip and (b) location of linear strip for $n_o = 20, 30, 40, 50, 60$ provided that background clutter type is Hardcore$(100, 6)$. Notice that vertical axes are differently scaled.

the clutter pattern is Hardcore$(100, 7)$ (see Figure 4.4), then the traversal length is maximized when $n_o = 40$ and linear width $\ell = 15$. Moreover, as we have already observed, the linear strip should locate closer to the target point. And, finally when the clutter pattern is Hardcore$(100, 8)$ the traversal length is again maximized for the true obstacle number level $n_o = 40$ with width $\ell = 5$, and locating at the bottom in the study window (see Figure 4.5).

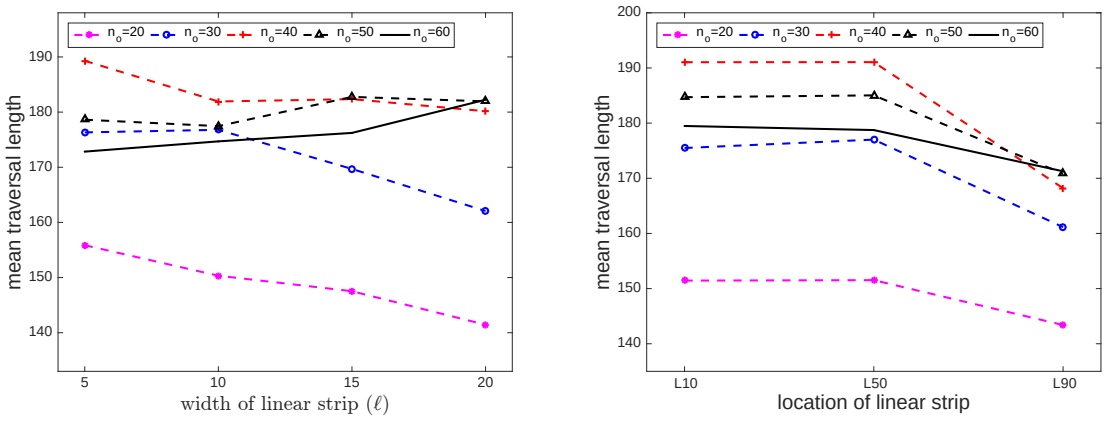


(a) (b)

Figure 4.4: Mean traversal length versus (a) width ℓ of linear strip and (b) location of linear strip for $n_o = 20, 30, 40, 50, 60$ provided that background clutter type is Hardcore$(100, 7)$. Notice that vertical axes are differently scaled.

4.2.2 V-Shaped Window

We will use similar setting as in Section 4.2.1 where the true obstacle window form is V-shaped, and true obstacles are uniformly distributed inside the V-shaped form (Figure 4.6). Let the distance between two upper lips of V-shaped obstacle window be v, then we consider various v values ranging from 20 to 80 with an increment size of 20. Together with the v value, we also change the location of V-shaped obstacle form; close to target, at center, and close to source. Let V50, V70, and V90 be V-shaped obstacle forms locating at the bottom, center, and at the top of the study window.



(a) (b)

Figure 4.5: Mean traversal length versus (a) width ℓ of linear strip and (b) location of linear strip for $n_o = 20, 30, 40, 50, 60$ provided that background clutter type is Hardcore$(100, 8)$. Notice that vertical axes are differently scaled.

V50 refers to leftmost upper corner of V-shaped obstacle that follows the points $(10, 50), (10, 40), (50, 10), (90, 40), (90, 50), (50, 20)$ in counter-clock wise direction and forms the desired V-shaped polygon. V70 and V90 are defined analogously. In the work of Aksakalli and Ceyhan (2012), the longest traversal length occurred when the background clutter type was of Hardcore$(100, 5)$ point process with $n_o = 50$ true obstacles uniformly distributed inside the V-shaped obstacle form with $v = 80$ locating at the top of working space (i.e., close to target).

When the background clutter pattern is from Hardcore$(100, 5)$ we observe that the traversal length tends to increase as the distance v between upper lips of V-shaped obstacle form increases for n_o greater than or equal to 40 (Figure 4.7(a)), and maximized for true obstacle number level $n_o = 50$ together with the location of obstacle form at the top of study window (Figure 4.7(b)). We will also investigate the trend of traversal length when the background clutter changes as well.

Since, the trends in mean traversal length under the background clutter pattern Hardcore$(100, 5)$, Hardcore$(100, 6)$, and Hardcore$(100, 7)$ are similar, we only provide

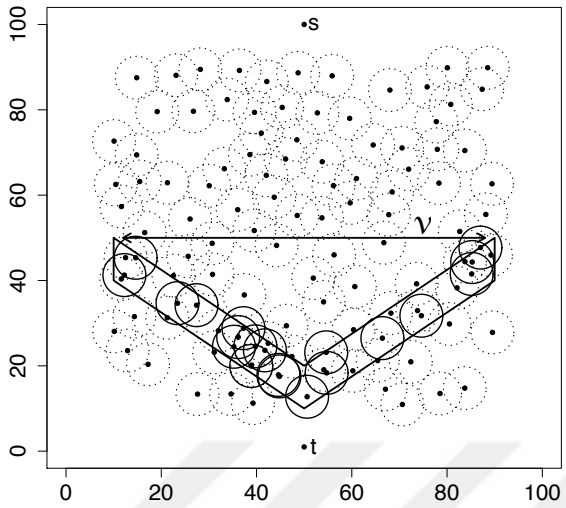
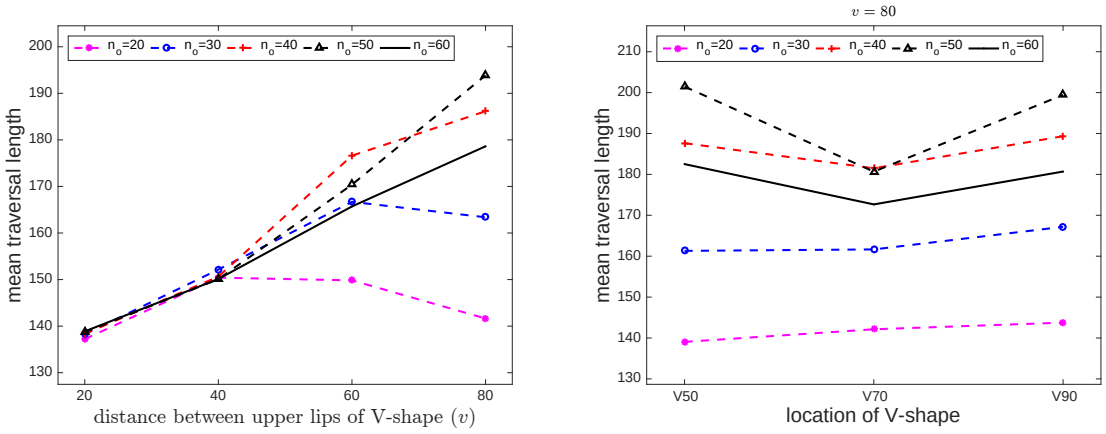


Figure 4.6: The background clutter pattern is from the Hardcore$(100, 5)$ (dashed circles) process and $n_o = 20$ true obstacles are uniformly distributed inside the V-shaped obstacle window with distance $v = 80$ between two upper lips (solid circles).

the plot for the case Hardcore$(100, 5)$ (Figure 4.8). In Figure 4.8, contrary to other cases, the traversal length tends to be larger for obstacle number level $n_o = 40$ than the obstacle number level $n_o = 50$. Moreover, in all cases the traversal length tends to be larger when the obstacle form is located at the top of the study window (i.e., location of V-shaped obstacle form is V90).

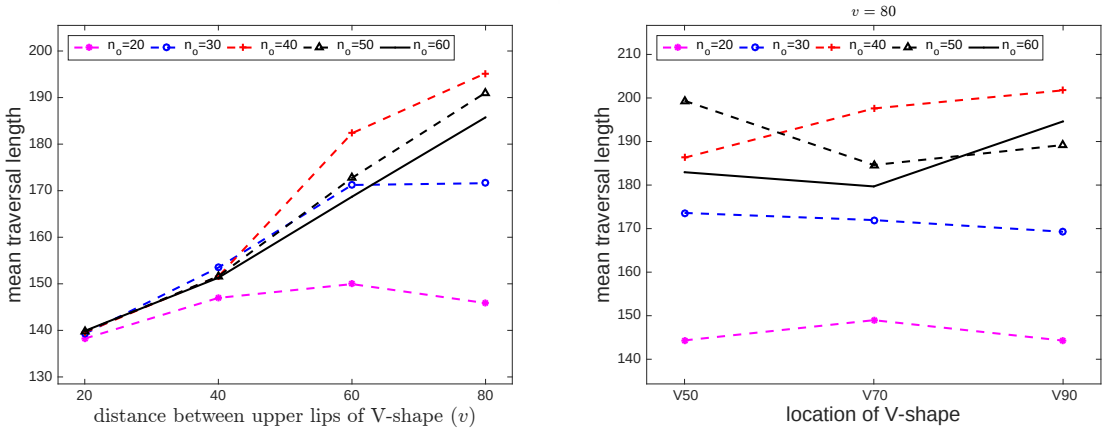
In Table 4.1, the maximum traversal length changes with respect to the background obstacle pattern. For Hardcore clutter patterns with $d = 5, 6, 7$, the mean traversal length is maximized for the obstacle number level $n_o = 50$, whereas for the clutter pattern with $d = 8$, the mean traversal length is maximized for the obstacle number level $n_o = 40$.

We observed that the mean traversal length tends to be larger when the V-shaped obstacle window form is located close to the source point. Let w be the distance between the vertex of V-shaped obstacle window form and the coordinate point $(50, 10)$. We will investigate the trend of mean traversal length of NAVA when the upper two lips of the obstacle form are fixed as shown in Figure 4.9, but the distance w changes



(a) (b)

Figure 4.7: Mean traversal length versus (a) v value and (b) location of V-shaped window for $n_o = 20, 30, 40, 50, 60$ together with $v = 80$ provided that the background clutter type is Hardcore$(100, 5)$. Notice that vertical axes are differently scaled.



(a) (b)

Figure 4.8: Mean traversal length versus (a) v value and (b) location of V-shaped window for $n_o = 20, 30, 40, 50, 60$ with $v = 80$ provided that background clutter type is Hardcore$(100, 8)$. Notice that vertical axes are differently scaled.

	$n_o = 40; n_o = 50$		
$v = 80$	V50	V70	V90
HC($n_c = 100, d = 5$)	187.63; **201.47**	181.49; 180.67	189.33; 199.44
HC($n_c = 100, d = 6$)	189.62; **193.29**	190.58; 192.13	181.12; 185.06
HC($n_c = 100, d = 7$)	183.24; 192.00	191.80; **199.06**	194.55; 186.49
HC($n_c = 100, d = 8$)	186.28; 199.35	197.59; 184.60	**201.75**; 189.19

Table 4.1: Comparison of mean traversal lengths for obstacle number levels $n_o = 40$ and $n_o = 50$.

from 0 to 70 with an increment size of 10 units. Note that the total area of V-shaped obstacle window form still remains to be 80 units as w varies (for consistency the area of all obstacle forms are 800 units).

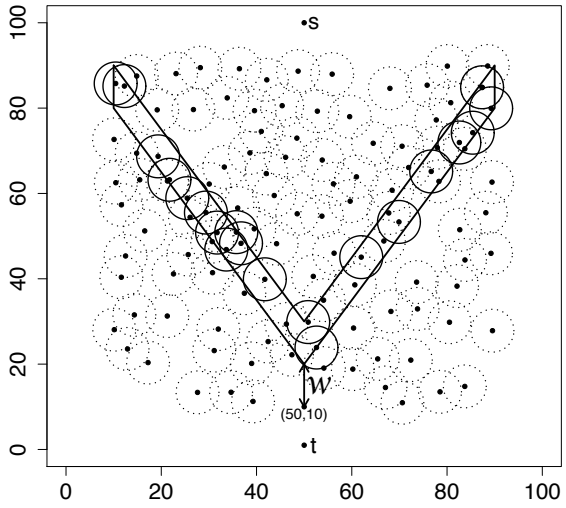


Figure 4.9: The background clutter pattern is from the Hardcore$(100, 5)$ (dashed circles) process and $n_o = 20$ true obstacles (solid circles) are uniformly distributed inside the V-shaped obstacle window with distance $w = 10$ between the vertex of the obstacle from and the point $(50, 10)$.

In Figure 4.10, the mean traversal length is maximized when $w = 10$ (or $w = 20$) for true obstacle number level $n_o = 60$. Moreover, for each true obstacle number level we have a concave down trend for mean traversal lengths, and observe that when $w = 70$ then the obstacle form takes the shape of linear strip which we have already discussed in previous sections.

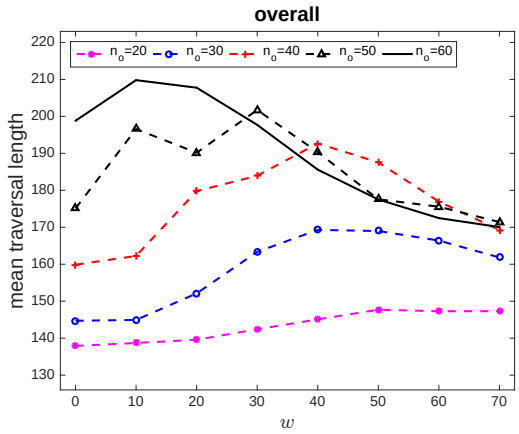


Figure 4.10: Overall mean traversal length versus w values for $n_o = 20, 30, 40, 50, 60$.

4.2.3 Semicircular Window

We will use similar setting as in Section 4.2.1 but the true obstacle window form is semicircular, and true obstacles are uniformly distributed inside the Semicircle form (Figure 4.11). We also change the location of semicircular obstacle form; close to target, at center, and close to source. Let SC50, SC70, and SC90 be semicircular obstacle window forms locating at the bottom, center, and at the top of the study window. SC50 refers to the center of semicircle where outer radius is 40 units and inner radius is 33 units. Since, we want all obstacle forms to have almost equal areas (for consistency), inner and outer radii of semicircle are chosen accordingly. That is to say, linear strip of width 10 units has area of 800 units, V-shaped with $v = 80$ has area of 800 units, and semicircle with given radii has area of 800 units. SC70 and SC90 are defined analogously.

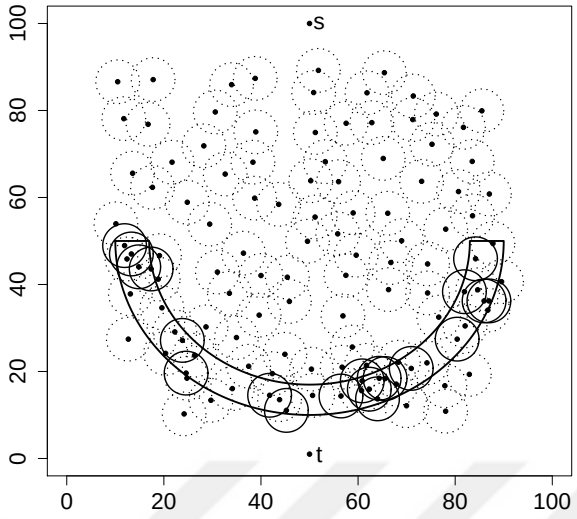
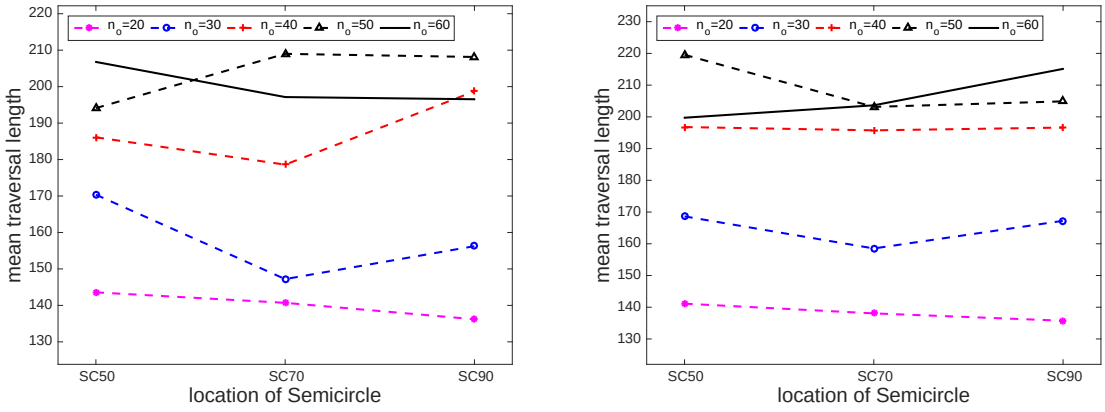


Figure 4.11: The background clutter pattern is from the Hardcore$(100, 5)$ (dashed circles) process and $n_o = 20$ true obstacles are uniformly distributed inside the semicircular obstacle window form (solid circles).

When the background clutter pattern is from Hardcore$(100, 5)$ we observe that the traversal length tends to be larger for true obstacle levels $n_o = 50$ and $n_o = 60$ regardless of the location of semicircular obstacle form (Figure 4.12). Since, the trends in mean traversal length under the background clutter pattern Hardcore$(100, 5)$, Hardcore$(100, 6)$, Hardcore$(100, 7)$, and Hardcore$(100, 8)$ are similar, we only provide the plot for the cases Hardcore$(100, 5)$ and Hardcore$(100, 8)$ (Figure 4.12), and maximum is attained in the case of background clutter type Hardcore$(100, 6)$ at true obstacle level $n_o = 60$ (Table 4.2).

In Figure 4.12(a), the mean traversal length tends to be larger for obstacle number level $n_o = 50$ than $n_o = 60$, whereas in Figure 4.12(b) we observe the reverse inclusion. Moreover, in Table 4.2 we observe that traversal lengths are larger than that in Table 4.1, i.e., the semicircular obstacle window form yields higher traversal length compared to the V-shaped obstacle window form.

We observed that the mean traversal length tends to be larger when the semicircular obstacle form is located close to the source point. To generalize this idea, let u



(a) (b)

Figure 4.12: Mean traversal length versus location of semicircular obstacle window form for $n_o = 20, 30, 40, 50, 60$ provided that the background clutter type is (a) Hardcore$(100, 5)$ and (b) Hardcore$(100, 8)$.

	$n_o = 50;\ n_o = 60$		
	SC50	SC70	SC90
HC$(n_c = 100, d = 5)$	194.11; 206.77	**208.98**; 197.14	**208.13**; 196.53
HC$(n_c = 100, d = 6)$	190.99; 208.06	205.22; **222.92**	203.68; 192.02
HC$(n_c = 100, d = 7)$	211.14; **211.70**	202.14; 206.68	203.40; 195.53
HC$(n_c = 100, d = 8)$	**219.55**; 199.72	203.14; 203.63	204.88; 215.12

Table 4.2: Comparison of mean traversal lengths for obstacle number levels $n_o = 50$ and $n_o = 60$.

be the distance between the vertex of semicircular obstacle form and the coordinate point $(50, 10)$. We will investigate the trend of mean traversal length of NAVA when the upper two lips of the obstacle form are fixed as shown in Figure 4.13, but the distance u changes from 0 to 70 with an increment size of 10 units. Note that the

total area of elliptical window obstacle form still remains to be 800 units as u varies (for consistency the area of all obstacle forms are 800 units). Note that when $u = 40$, the elliptical obstacle form turns out to be the semicircular obstacle form.

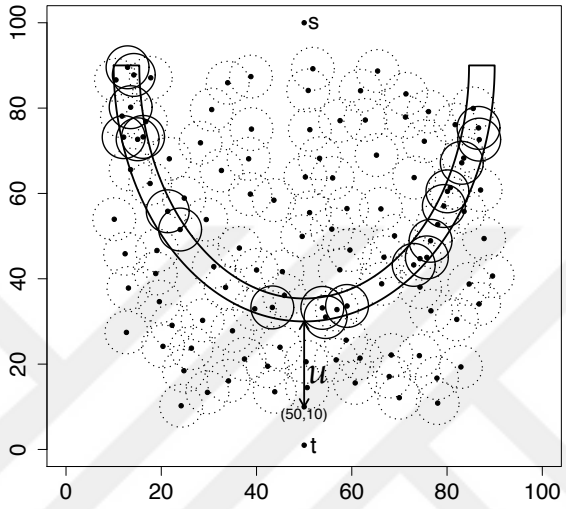


Figure 4.13: The background clutter pattern is from the Hardcore$(100, 5)$ (dashed circles) process and $n_o = 20$ true obstacles (solid circles) are uniformly distributed inside the elliptical obstacle with distance $u = 20$ between the vertex of the obstacle from and the point $(50, 10)$.

In Figure 4.14, the mean traversal length is maximized when $u = 20$ for true obstacle number level $n_o = 60$. Moreover, for each true obstacle number level we have a concave down trend for mean traversal lengths.

4.3 Placing True Obstacles in a Regular Fashion

We will use similar setting as in Section 4.2.1 where obstacle forms will still remain same, but the true obstacles inside obstacle forms will be placed as regular as possible. To achieve this, we will use Hardcore(n_o, d_2) point pattern for true obstacle pattern inside obstacle forms with parameters $n_o = 20, 30, 40$ and $d_2 = 5, 6, 7, 8$(Technical Report 4). Exploiting results from previous sections (Sections 4.2.1, 4.2.2, 4.2.3); for

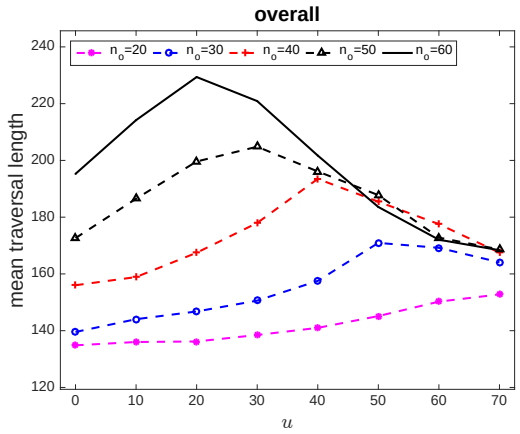


Figure 4.14: Overall mean traversal length versus u values for $n_o = 20, 30, 40, 50, 60$.

linear obstacle form we will use linear strip window locating close to target (L10) with width $\ell = 10$, for V-shaped obstacle window for we will use V-shaped form locating close to source point (V90) with distance between its upper lips equal to $v = 80$, and for semicircular obstacle window form we will use semicircle form locating close to the source point (SC90).

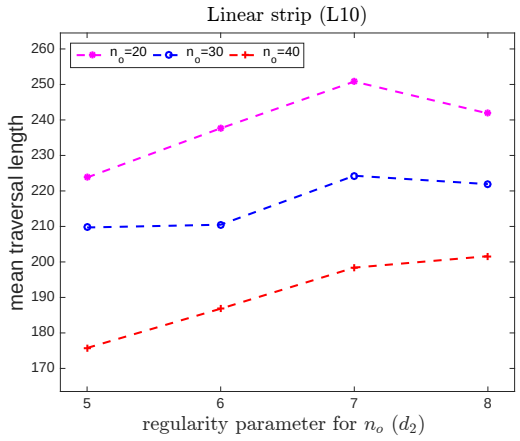


Figure 4.15: Mean traversal length versus d_2 values (regularity parameter for n_o) for $n_o = 20, 30, 40$ provided that the background clutter type is Hardcore (overall average of Hardcore$(n_c = 100, d)$ with $d = 5, 6, 7, 8$).

In Figure 4.15, we observe that the mean traversal length tends to be larger for true obstacle level $n_o = 20$. In Section 4.2.1, the largest traversal length was around 190 units and this value was achieved when $n_o = 40$ true obstacles uniformly distributed inside the linear obstacle form. Whereas, when true obstacles are distributed regularly, the mean traversal length is maximized for true obstacle number level $n_o = 20$, and it has value around 250 units which is considerably higher than 190 units.

	$n_o = 20$			
	$d_2 = 5$	$d_2 = 6$	$d_2 = 7$	$d_2 = 8$
HC($n_c = 100, d = 5$)	221.68	244.66	**252.29**	238.58
HC($n_c = 100, d = 6$)	230.18	227.13	241.98	**243.93**
HC($n_c = 100, d = 7$)	219.02	237.35	**246.45**	243.80
HC($n_c = 100, d = 8$)	224.24	241.45	**262.49**	241.01

Table 4.3: Mean traversal length versus d_2 values for $n_o = 20$ inside L10 obstacle form with background clutter type Hardcore($n_c = 100, d$) where $d = 5, 6, 7, 8$.

In Table 4.3, we provide mean traversal lengths separately for each background clutter type together with the regularity parameter for true obstacle number n_o (i.e., true obstacles are distributed regularly inside linear strip using Hardcore(n_o, d_2) point pattern). Hence, the mean traversal length is maximized (262.49 units) when the background clutter type is from Hardcore($n_c = 100, d = 8$) together with $n_o = 20$ true obstacles are distributed regularly inside the linear strip using Hardcore($n_o = 20, d_2 = 7$) point pattern.

In Figure 4.16, we observe that the mean traversal length tends to be larger for true obstacle level $n_o = 30$. In Section 4.2.2, the largest traversal length was around 200 units (Table 4.1) and this value was achieved when $n_o = 40$ (or $n_o = 50$) true obstacles uniformly distributed inside the V-shaped obstacle window form. Whereas, when true obstacles are distributed regularly, the mean traversal length is maximized for true obstacle number level $n_o = 30$, and it has value around 260 units which is

substantially higher than 200 units.

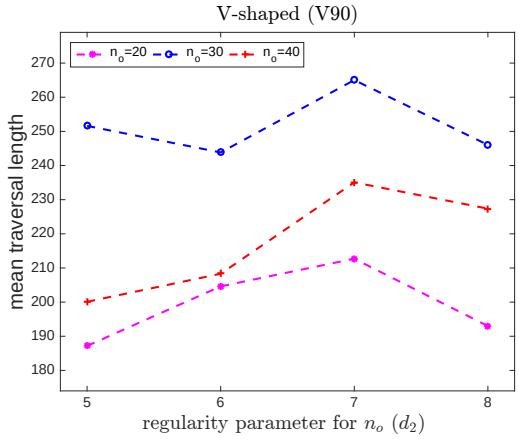


Figure 4.16: Mean traversal length versus d_2 values (regularity parameter for n_o) for $n_o = 20, 30, 40$ provided that the background clutter type is Hardcore (overall average of Hardcore($n_c = 100, d$) with $d = 5, 6, 7, 8$).

In Table 4.4, we provide mean traversal lengths separately for each background clutter type together with the regularity parameter for true obstacle number n_o (i.e., true obstacles are distributed regularly inside V-shaped obstacle using Hardcore(n_o, d_2) point pattern). Hence, the mean traversal length is maximized (277.95 units) when the background clutter type is from Hardcore($n_c = 100, d = 8$) together with $n_o = 30$ true obstacles are distributed regularly inside the V-shaped obstacle form using Hardcore($n_o = 30, d_2 = 7$) point pattern.

On the other hand, from the construction shown in Figure 4.9 we observed that the mean traversal length tends to be larger when $w = 10$ (Figure 4.10). We investigated the trend of mean traversal length when true obstacles are placed regularly inside the new V-shaped obstacle form with $w = 10$, and we observe that the mean traversal length is maximized for true obstacle number level $n_o = 30$ with regularity parameter $d_2 = 6$ (i.e., 405 units, Figure 4.17).

In Figure 4.18, we observe that the mean traversal length tends to be larger for true obstacle levels $n_o = 20$ or $n_o = 30$. In Section 4.2.3, the largest traversal length

	$n_o = 30$			
	$d_2 = 5$	$d_2 = 6$	$d_2 = 7$	$d_2 = 8$
HC$(n_c = 100, d = 5)$	**260.76**	239.73	252.76	251.17
HC$(n_c = 100, d = 6)$	235.37	251.42	**262.10**	234.58
HC$(n_c = 100, d = 7)$	250.76	235.98	**267.27**	242.75
HC$(n_c = 100, d = 8)$	259.49	248.21	**277.95**	255.67

Table 4.4: Mean traversal length versus d_2 values for $n_o = 30$ inside V90 obstacle form with background clutter type Hardcore$(n_c = 100, d)$ where $d = 5, 6, 7, 8$.

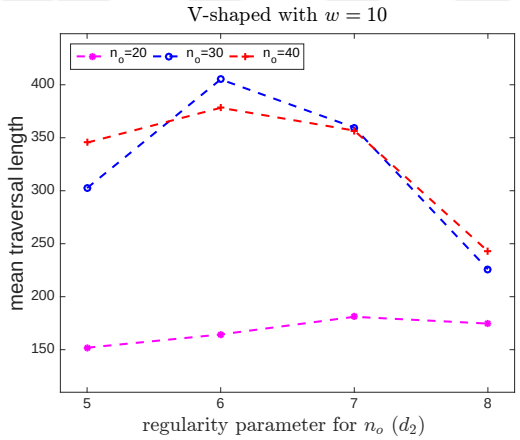


Figure 4.17: Mean traversal length versus d_2 values (regularity parameter for n_o) for $n_o = 20, 30, 40$ provided that the background clutter type is Hardcore (overall average of Hardcore$(n_c = 100, d)$ with $d = 5, 6, 7, 8$).

was around 230 units (Table 4.2) and this value was achieved when $n_o = 50$ (or $n_o = 60$) true obstacles uniformly distributed inside the semicircular obstacle form. Whereas, when true obstacles are distributed regularly, the mean traversal length is maximized for true obstacle number level $n_o = 30$, and it has value around 316 units which is noticeably higher than 230 units.

In Table 4.5, we provide mean traversal lengths separately for each background

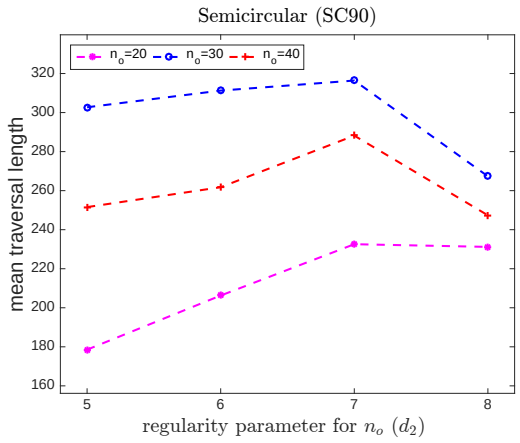


Figure 4.18: Mean traversal length versus d_2 values (regularity parameter for n_o) for $n_o = 20, 30, 40$ provided that the background clutter type is Hardcore (overall average of Hardcore($n_c = 100, d$) with $d = 5, 6, 7, 8$).

clutter type together with the regularity parameter for true obstacle number n_o (i.e., true obstacles are distributed regularly inside semicircular obstacle form using Hardcore(n_o, d_2) point pattern). Hence, the mean traversal length is maximized (328 units) when the background clutter type is from Hardcore($n_c = 100, d = 5$) together with $n_o = 30$ true obstacles are distributed regularly inside the semicircular obstacle form using Hardcore($n_o = 30, d_2 = 5$) point pattern.

	$n_o = 30$			
	$d_2 = 5$	$d_2 = 6$	$d_2 = 7$	$d_2 = 8$
HC($n_c = 100, d = 5$)	319.38	315.64	**328.40**	269.65
HC($n_c = 100, d = 6$)	294.70	**305.35**	300.56	274.10
HC($n_c = 100, d = 7$)	293.41	298.99	**314.12**	250.69
HC($n_c = 100, d = 8$)	303.33	**325.05**	322.52	274.93

Table 4.5: Mean traversal length versus d_2 values for $n_o = 30$ inside SC90 obstacle form with background clutter type Hardcore($n_c = 100, d$) where $d = 5, 6, 7, 8$.

On the other hand, from the construction shown in Figure 4.13 we observed that the mean traversal length tends to be larger when $u = 20$ (Figure 4.14). We investigated the trend of mean traversal length when true obstacles are placed regularly inside the elliptical obstacle form with $u = 20$, and we observe that the mean traversal length is maximized for true obstacle number level $n_o = 30$ with regularity parameter $d_2 = 6$ (i.e., 465 units, Figure 4.19).

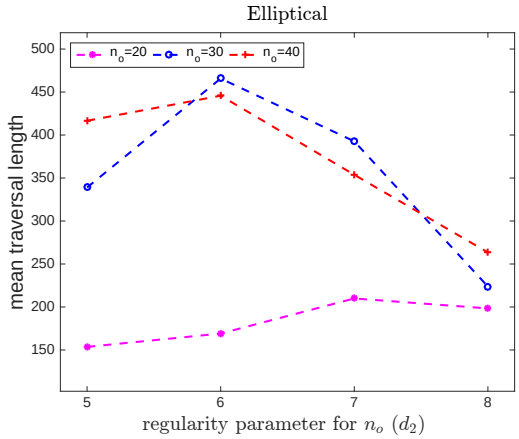


Figure 4.19: Mean traversal length versus d_2 values (regularity parameter for n_o) for $n_o = 20, 30, 40$ provided that the background clutter type is Hardcore (overall average of Hardcore$(n_c = 100, d)$ with $d = 5, 6, 7, 8$).

4.4 Voronoi and Delaunay Tessellations

Given a set of finite (two or more) points in the Euclidean plane, we associate all locations in that space with the closest member(s) of the point set with respect to the Euclidean distance. The result is a tessellation of the plane into a set of the regions associated with members of the point set. We call this tessellation the *Voronoi diagram* generated by the point set, and regions constituting the Voronoi diagram *Voronoi polygons* (Okabe et al. (2000)). Formally,

Definition 4.4.1 *(Voronoi Tessellation) Let* $P = \{p_1, p_2, \ldots, p_n\} \subset \mathbb{R}^2$ *and* $I_n =$

$\{1, 2, \ldots, n\}$. *We call the region given by*

$$V(p_i) = \{p \in \mathbb{R}^2 \mid \|p - p_i\| \leq \|p - p_j\| \text{ for } j \neq i, j \in I_n\}$$

the Voronoi polygon of p_i*, and the set given by*

$$V(P) = \{V(p_1), V(p_2), \ldots, V(p_n)\}$$

the Voronoi diagram of P.

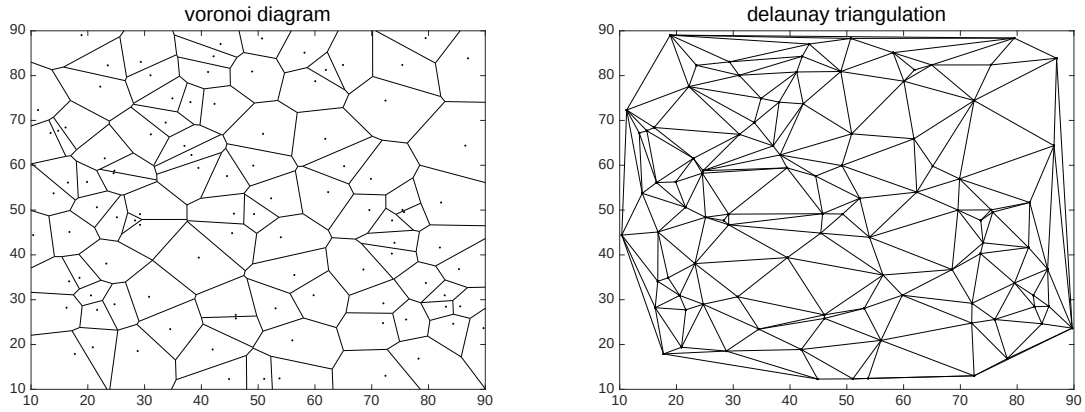
Similarly, as a planar graph has its dual graph, the dual tessellation of a Voronoi diagram is called the Delaunay triangulation. A Delaunay triangulation of $D(P)$ is a triangulation of P such that no point in P is inside the circumcircle of any triangle T in $D(P)$. If n_v is the number of Voronoi vertices of Voronoi diagram $V(P)$, then the set $D(P) = \{T_1, T_2, \ldots, T_{n_v}\}$ satisfies

$$\text{convex hull (CH) of } P = \cup_{i=1}^{n_v} T_i,$$

such that $[T_i \setminus \partial T_i] \cap [T_j \setminus \partial T_j] = \emptyset$, $j \neq i, j \in I_{n_v}$. For illustration, the Voronoi and Delaunay tessellations of 100 randomly selected points from $\mathbb{R}^2$ are presented in Figure 4.20.

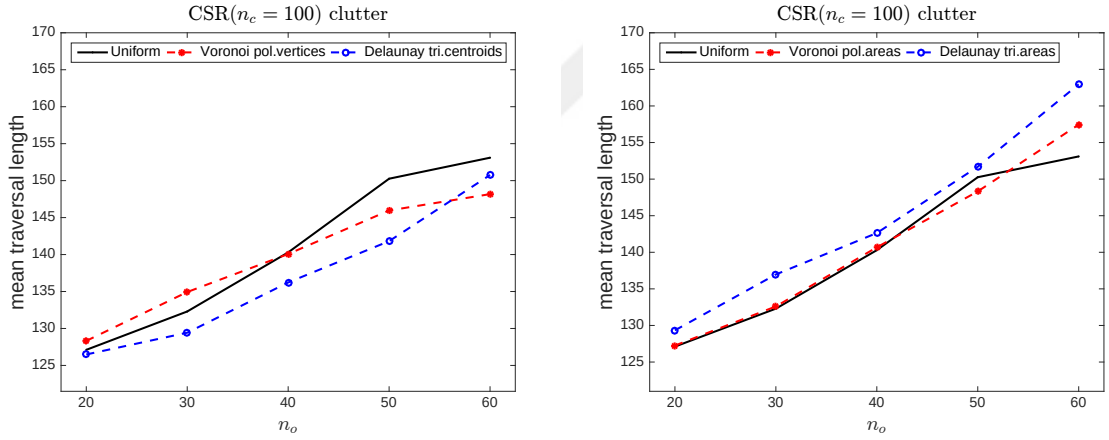
4.4.1 Results and Analysis

In our experimental setting, there are mainly two cases that we investigate. In the first case, the number of uniformly sampled false obstacles (n_c) is fixed to 100, and the number of true obstacles (n_o) is considered to be $20, 30, 40, 50$, and 60. For the fixed number of false obstacles ($n_c = 100$), Voronoi and Delaunay tessellation is carried out, and then true obstacles from the set $\{20, 30, \ldots, 60\}$ are placed around the vertices of Voronoi polygons and the centroids of Delaunay triangulation, respectively. Moreover, true obstacles are also placed proportional to areas of Voronoi polygons and Delaunay triangles. These four types of settings are compared to the case where true obstacles are distributed uniformly inside the study window. In Figure 4.21(a), we observe that as the number of obstacles increase the average mean traversal length increases as well.



(a) (b)

Figure 4.20: (a) Voronoi diagram and (b) Delaunay triangulation of 100 sample points.

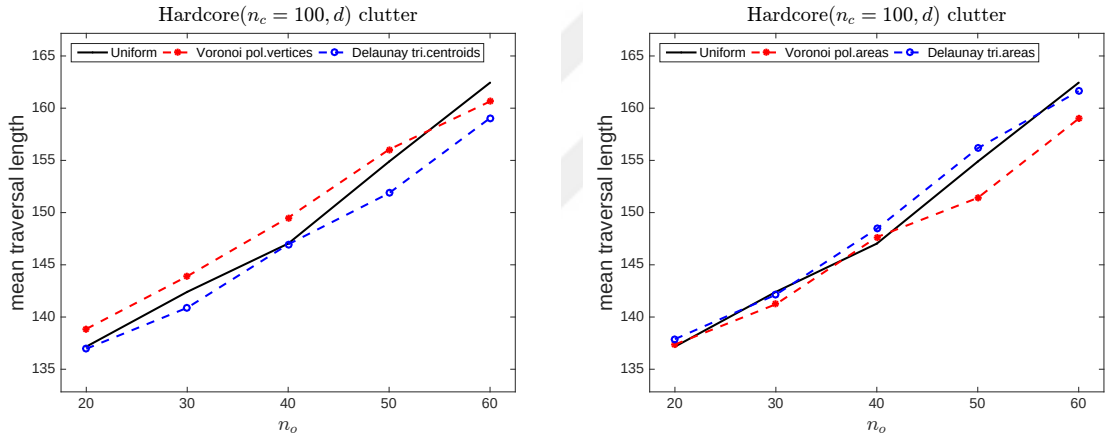


(a) (b)

Figure 4.21: For the CSR background clutter type, the mean traversal length (a) when true obstacles are placed around vertices and centroids of tessellations, and (b) when true obstacles are placed proportional to tile areas of tessellations.

When true obstacles are uniformly distributed the mean traversal length tends to be larger than the mean traversal length when true obstacles placed around vertices of Voronoi polygons or around centroids of Delaunay triangles. So, from OPA's point of

view, in this case knowing the exact places of obstacles does not give an advantage in aiming to maximize the total traversal length of NAVA. On the other hand, in Figure 4.21(b) we observe that when obstacles placed proportional to tessellation areas, then it tends to yield higher traversal length than the case where obstacles are distributed uniformly. Especially, there is a noticeable increase in mean traversal length when obstacles are placed inside Delaunay triangles proportional to their areas. Note that in Figure 4.21, the maximum mean traversal length is less than 165 units. However, for the CSR background clutter type together when true obstacles are uniformly placed inside the V-shaped obstacle form, then the mean traversal lengh can achieve as much as 178 units (Aksakalli and Ceyhan (2012) and Technical Report 4).



(a) (b)

Figure 4.22: For the Hardcore($n_c = 100, d$) background clutter type (combined all $d = 5, 6, 7, 8$ cases), the mean traversal length (a) when true obstacles are placed around vertices and centroids of tessellations, and (b) when true obstacles are placed proportional to tile areas of tessellations.

In the second case, we will repeat the first case with slight modification. In this case, false obstacles are placed as regular as possible using the hardcore point pattern and the rest of the setting is kept similar. For the regular background clutter pattern we use the Hardcore($n_c = 100, d$) point process where $d = 5, 6, 7, 8$ as in

Technical Report 4. In Figure 4.22, we observe similar trends as in Figure 4.21. Mean traversal length in Figure 4.21(a) tends to be smaller than in Figure 4.22(a) since in the previous case the background clutter type was from CSR and in the latter case it was from Hardcore (regular). When true obstacles are placed proportional to the tile areas of tessellations, on the average the mean traversal length for Delaunay case tends to be higher than the uniform case and for Voronoi case it tends to be smaller than the uniform case. In either cases, note that the maximum average traversal length is less than 165 units. But, when the background clutter type is regular (from hardcore point process) and true obstacles are uniformly distributed inside V-shaped obstacle form, then the mean traversal length tends to have around 190 units (Aksakalli and Ceyhan, 2012). Indeed, using the same number of false and true obstacles (but different obstacle forms) as in this problem setting, we observed that the mean traversal length can be as high as 450 units (Section 4.2.3).

4.5 Summary

In the OPD problem, on the behalf of OPA, we wish the traversal length of NAVA to be as larger as possible. To achieve this, we choose the clutter pattern as regular as possible i.e., the background clutter pattern is Hardcore(n_c, d) where n_c is the number of false obstacles, and d represents the distance such that there is no pairwise disk-shaped obstacles closer than d units (see Chapter 2). Together with this setting, true obstacles are uniformly distributed inside various obstacle forms such as linear strip, V-shaped, and semicircular obstacle window forms. In linear obstacle window form, the traversal length tends to be larger for true obstacle number level $n_o = 40$ and that linear strip is locating at the bottom of study window (i.e., close to target point). In V-shaped obstacle form, the traversal length tends to be larger for obstacle number levels $n_o = 40$ or 50 depending on the background clutter type (i.e., Hardcore(n_c, d) with $d = 5, 6, 7, 8$). In semicircular obstacle form, the traversal length yields higher values for true obstacle number levels $n_0 = 50$ or 60. Note that, to attain consistent results the total area covered by obstacle forms are fixed to 800 units. Based on our

Monte Carlo simulations, we observe that in overall comparison the traversal length is maximized when obstacle form is the elliptical obstacle window locating close to the source point.

Next, we investigated the case when the background clutter type is again of Hardcore type but true obstacles are distributed as regular as possible rather than uniform. One of advantages of studying this situation is that the traversal length might be maximized using fewer true obstacles. According to our results, when the background clutter type is from Hardcore($n_c = 100, d = 8$) together with $n_o = 20$ true obstacles distributed regularly inside the linear strip obstacle form using Hardcore($n_o = 20, d_2 = 7$) point pattern, the mean traversal length is 262.49 units which yields higher value than the case when $n_o = 40$ true obstacles are distributed uniformly inside the linear strip (i.e., approximately 190 units). Notice that, in each of these two cases the background clutter patterns remains the same.

Moreover, working with the same background clutter types but changing true obstacle from linear strip to V-shaped obstacle form (specifically V90), we obtain even higher mean traversal lengths. The mean traversal length attains the value of 277.95 units when background clutter type is from Hardcore($n_c = 100, d = 8$) together with 30 true obstacles are distributed regularly inside V-shaped obstacle form using Hardcore($n_o = 30, d_2 = 30$) point pattern. On the other hand, recall that when true obstacles are distributed uniformly inside V-shaped obstacle form, the longest mean traversal was of 201.75 units for true obstacle number level $n_o = 50$. Thus, again the traversal length is maximized for V-shaped obstacle form using fewer true obstacles. When we consider the V-shaped obstacle form as shown in Figure 4.9, the mean traversal length tends to be larger compared to standard V90 obstacle form. The mean traversal length is about 210 when $n_o = 60$ true obstacles are uniformly distributed (Figure 4.10), and 405 units when $n_o = 30$ true obstacles are regularly distributed (Figure 4.17). So, in order to maximize the mean traversal length we recommend using the V-shaped obstacle form as shown in Figure 4.9 among V-shaped obstacle forms.

Finally, in Sections 4.2.1, 4.2.2, 4.2.3 we observed that among obstacle forms the elliptical obstacle form yields the longest mean traversal when true obstacles are uniformly distributed inside obstacle window forms (i.e., linear strip, V-shaped, semicircular, and elliptical) for various background clutter types from Hardcore($n_c = 100, d$) with $d = 5, 6, 7, 8$. And, in the case of Semicircle obstacle form the longest mean traversal length has value of 222.92 units when clutter type is from Hardcore($n_c = 100, d = 6$) together with 60 true obstacles distributed uniformly inside semicircular obstacle form (SC70). But, when true obstacles are distributed regularly inside semicircular shape (SC90) using Hardcore($n_o = 30, d_2 = 5$) the mean traversal length attains the longest value of 328.40 units (which is considerably higher than 222.92 units). When we consider the elliptical obstacle form as shown in Figure 4.13, the mean traversal length tends to be larger compared to standard SC90 obstacle form. The mean traversal length is about 230 when $n_o = 60$ true obstacles are uniformly distributed (Figure 4.14), and 465 units when $n_o = 30$ true obstacles are regularly distributed (Figure 4.19). Hence, the mean traversal length is maximized when 30 true obstacles are distributed regularly (i.e., Hardcore($n_o = 30, d_2 = 6$)) inside the elliptical obstacle form as shown in Figure 4.13 ($u = 20$) together with Hardcore($n_c = 100, d$) background clutter type for $d = 5, 6, 7, 8$.

On the other hand, we observed that knowing the exact locations of obstacles does not provide significant gain for NAVA more advantage than knowing only the distribution of obstacles. Placing true obstacles proportional to areas of Voronoi polygons or Delaunay triangles, which are based on the allocation of clutters, seems to have less effect on maximizing the total traversal length of NAVA. The mean traversal length with respect to tessellations results in slightly higher than the uniformness case, so we do not recommend using the tessellations so as to maximize the traversal length.

Chapter 5

HEURISTIC ALGORITHMS FOR THE OPD PROBLEM

5.1 Introduction

We will present some current algorithms used in solving the discretized SOS problem (D-SOSP). Taking the advantage of those algorithms' weaknesses, we mainly focus on developing new algorithms and improving them, and generalize the family of such algorithms. By Monte Carlo simulations, we will compare the empirical performances of the present and new algorithms. Moreover, we will adapt the algorithms and the penalty functions to discrete SOSP (i.e., a graph version) together with CTP where sensing is allowed.

5.2 Algorithms for NAVA

Consider the discretized version of SOS problem. Let G be a graph defined in Section 2.2 with vertex set V and edge set E. Also, suppose that $D = D_x$ is a disk with center $x \in X = X_C \cup X_O$ with given radius $r > 0$. Given a disambiguation problem instance, our goal is to find an s, t walk in $G = (V, E)$ such that the total traversal length is minimized with a dynamic disambiguation cost. Here, a *walk* is an alternating sequence of vertices and edges, starting and ending at a vertex, in which each edge is adjacent in the sequence to its two endpoints. And, a *path* is a walk without repeated vertices or edges. NAVA can disambiguate only when situated on the boundary of a disk, i.e, when situated at the point of the edge intersecting the disk. NAVA has an option to learn whether D is traversable or not, at a cost c added to the total length of the traversal (throughout the rest of article we assume that all heuristic algorithms always disambiguate in a greedy manner). In the RD algorithm, prior to the NAVA's

traversal all edges are equipped with the weight function $w(e)$,

$$w(e) = \ell(e) + \frac{1}{2}\sum_{i=1}^{|X|} \mathbf{1}_{\{e\cap D_i\neq\emptyset\}} \times \left(\frac{c}{1-p_i}\right)$$

where $\ell(e)$ is the length of an edge e in the Euclidean sense and $\mathbf{1}_A$ is the indicator function for event A. In the RD algorithm, we simplify the notation and prefer the following,

$$w_{RD}(c,p) = \ell(e) + \frac{1}{2}\sum_{i=1}^{|X|} \mathbf{1}_{\{e\cap D_i\neq\emptyset\}} \times \mathcal{F}_{RD}(c,p_i)$$

where we denote the penalty function by $\mathcal{F}_{RD}(c,p) = \frac{c}{1-p}$ associated with the RD protocol (Aksakalli et al., 2011). Next, NAVA determines the shortest s,t path in G with respect to the edge weights $w_{RD}(c,p)$ (i.e., the shortest path is found by using the Dijkstra's algorithm (Cormen et al., 2009), and traverses this path until it encounters the first ambiguous edge $e = uv$ intersecting one or more disks D_i. Since the algorithm is greedy, NAVA uses its disambiguation option at vertex u and learns the actual status of the disk(s) D_i at cost c. If D_i is traversable (i.e., obstacle is false), then we remove D_i from the graph and continue the walk (i.e., reset the algorithm). If D_i is non-traversable (i.e., obstacle is true), then we set $p_i = 1$ and update the weights of the graph G, that is we delete all edges intersecting the disk(s) D_i (disambiguating an edge will consist of disambiguating all disks the edge enters, at a fixed cost per disk). NAVA replans the route from the vertex u until it reaches the target t by disambiguating and replanning as necessary. This protocol with associated weight function w_{RD} is the reset disambiguation (RD) Protocol of Aksakalli et al. (2011).

In the simulated risk (SR) algorithm of Fishkind et al. (2007), authors use the penalty function

$$\mathcal{F}_{SR}(\alpha,p) = \alpha\log\left(\frac{1}{1-p}\right)$$

where α is a constant. But, the main disadvantage of the SR algorithm is the "fine-tune" penalty term via the α parameter for improved performance. In Aksakalli et al.

(2011), authors show that the algorithm using the penalty function $\mathcal{F}_{RD}$ performs better than the algorithm using the penalty function $\mathcal{F}_{SR}$ and illustrate this by simulated experiments.

Our goal is to define a better performing algorithm compared to both RD and DT algorithms. Observe that when the disambiguation cost is zero or small, then the RD algorithm becomes useless because in the computation of $w_{RD}(c, p)$ we lose the probabilistic information and the disambiguation cost effects. To overcome this problem, the disambiguation cost should be additive in the penalty function. Also, unfortunately the RD algorithm fails to see the whole picture and may lead NAVA into a local trap. This argument is supported by the work of Aksakalli and Ceyhan (2012), where the mean traversal length of NAVA tends to be larger when the true obstacles are uniformly distributed inside a V-shaped obstacle form in an environment with uniformly distributed clutter (i.e., false obstacle) point pattern compared to other obstacle window types. To avoid such instances, we proposed the following penalty function

$$\mathcal{F}_P(c, p) = c + \frac{5p}{1 - p^{1-p}}$$

Based on our simulation results, we observe that an algorithm is better than the RD algorithm in the sense that the former algorithm gives smaller mean traversal length. We present and discuss the simulation results in the following sections.

The distance to termination (DT) algorithm of Aksakalli and Ari (2014) has outperformed all previous algorithms in many cases they considered. Aksakalli and Ari (2014) propose that any meaningful penalty function needs to be monotonically nondecreasing in c and p. This idea is also supported by work of Ye et al. (2011) reporting how the RD algorithm behaves when the sensor of NAVA changes from poor to perfect detection of true obstacles. Based on this observation, Aksakalli and Ari (2014) experimented with a large number of penalty functions with additive cost terms that are also monotonically nondecreasing in c and p. As a result, they come up with the

following penalty function

$$\mathcal{F}_{DT}(c,p) = c + \left(\frac{d_t}{1-p}\right)^{-\log(1-p)}$$

where d_t is the Euclidean distance from the center of disk D to the target point t. Observe that, the penalty function $\mathcal{F}_{DT}(c,p)$ grows exponentially faster than others when p values increase. This property of the DT algorithm distinguishes it from others. It takes almost no risk and prefers to disambiguate as few obstacles as possible on its way. A similar idea was also mentioned in the work of Bnaya et al. (2008, 2009), where sensing in CTP problem corresponds to a disambiguation option in the D-SOSP. They consider always sense (cost=0), never sense (cost=infinity), constant sense (cost=constant), and sense cost proportional to the distance from the current location to the starting point s. In traditional CTP, the zero-risk path from s to t is assumed to have very large value (i.e., using chopper rather than driving), and hence the navigating agent will not prefer it unless all edges in the given graph G are revealed to be blocked. Whereas, in the D-SOSP the study window is $[0,100]\times[0,100]$ and obstacles are distributed inside $[10,90]\times[10,90]$, so there is always a path from s to t circumnavigating obstacles. Due to the penalty function's characteristic, the DT algorithm is reluctant to disambiguate obstacles in a greedy manner and prefers not to take any risk because the zero-risk path always remains to be as the plausible choice and thus is taken for granted.

We also introduce an algorithm called the 'benchmark algorithm' for comparative purposes. In this algorithm, NAVA knows both the location and the status of all obstacles, but (admittedly) still disambiguates whenever runs into an obstacle in a greedy manner. That is to say, NAVA assigns the probabilities for true obstacles as 1 and for false obstacles as 0 before the traversal, and follows the usual RD algorithm's steps to find the traversal length. If an obstacle is clutter, then NAVA traverses over it and the disambiguation cost is added to the traversal length. However, in the benchmark algorithm, NAVA never disambiguates true obstacles because it knows the actual status of obstacles and its sensor already assigned 1 as the probability of being true obstacle. And, when $p = 1$ the value of $\mathcal{F}_{RD}(c,p)$ becomes ∞. Hence, this

benchmark algorithm provides the shortest possible traversal length that any similar heuristic algorithm can achieve. Moreover, let us denote the zero-risk algorithm by Z where no disambiguation take places in finding the shortest s, t walk.

Concerning the complexity of greedy algorithms, running time of finding the shortest path in a graph $G = (V, E)$ using the classical Dijkstra's algorithm is $O(|E| + |V| \log |V|)$ (Cormen et al., 2009). In the D-SOS problem, if the working window is $\Omega = [0, n] \times [0, n]$, then running time for finding the shortest path by the Dijkstra's algorithm is $O(n^2 \log(n))$. Analogously, the running time for finding the traversal length in the D-SOSP by using the RD algorithm is $O(n^2 \log(n)|X|r^4)$ (and hence, B, DT algorithms as well).

5.3 A Unifying Framework for $\mathcal{F}_{RD}$, $\mathcal{F}_P$, and $\mathcal{F}_{DT}$

We introduce a penalty function encompassing the previous ones. Consider the following function

$$\mathcal{F}_k(c, p) = c + \left(\frac{1}{1-p}\right)^k$$

where k is a nonnegative integer.

Our goal is to find a value k such that $\mathcal{F}_{RD}$ is sufficiently approximated by $\mathcal{F}_k$ for values of p in the interval $[0, 0.9]$. Here, it is reasonable to consider the interval $[0, 0.9]$ rather than $[0, 1]$ since as p gets closer to 1, all penalty functions (i.e., $\mathcal{F}_{RD}, \mathcal{F}_P, \mathcal{F}_{DT}$, and $\mathcal{F}_k$) get extremely large. One common way of efficient approximation is using the L^2 norm, i.e., find k such that $\|\mathcal{F}_{RD}(c, p) - \mathcal{F}_k(c, p)\|_{L^2}$ is minimized where the L^2 norm is defined as $\|\mathcal{F}_k(c, p)\|_{L^2} = \left(\int |\mathcal{F}_k(c, p)|^2 \, dp\right)^{1/2}$ with respect to the variable p. To achieve this, we fix the disambiguation cost $c > 0$, otherwise the k values may change as c changes. For example, when $c = 1$ then $\mathcal{F}_{RD}(c, p)$ is closely approximated by $\mathcal{F}_1(c, p)$, but when $c = 5$ then $\mathcal{F}_{RD}(c, p)$ is closely approximated by $\mathcal{F}_2(c, p)$. Thus, for any fixed value of c there always exist positive integers k such that any heuristic penalty function can be approximated by the penalty functions of the form $\mathcal{F}_k(c, p)$. To keep it simple throughout the article we let $c = 5$ (see Section 5.4), and in Figure

5.1(a) we observe that the range of $\mathcal{F}_{RD}(c,p)$ and $\mathcal{F}_2(c,p)$ coincides in the interval $[0, 0.9]$. Similarly, in Figure 5.1(b) we see that the range of $\mathcal{F}_P(c,p)$ and $\mathcal{F}_3(c,p)$ coincides in the interval $[0, 0.9]$. Based on our simulation results, the expression d_t in the penalty function $\mathcal{F}_{DT}$ can be also replaced by d_s (i.e., the distance from the center of disk D to the starting point s). This idea encourages us to replace d_t by $d(s,t)/2$ where $d(s,t)$ is the Euclidean distance between points s and t. In our experimental setting, $d(s,t)/2 = 50$ (see Section 5.4). Throughout this section, we use $d_t = 50$ in our plots to present the costs with $\mathcal{F}_{DT}$ penalty function, whereas in the actual DT algorithm we do not fix $d_t = 50$. Thus, in Figure 5.1(c) we observe that the range of $\mathcal{F}_{DT}(c,p)$ with $d_t = 50$ and $\mathcal{F}_5(c,p)$ coincides in the interval $[0, 0.7]$. The intervals $[0, 0.7]$ and $[0, 0.9]$ are sufficient to observe the behavior of the penalty functions. Since, for p values greater than 0.7 or 0.9 all penalty functions tend to increase exponentially. Among these penalty functions, the $\mathcal{F}_{RD}(c,p)$ has the longest rate of increase and the $\mathcal{F}_{DT}(c,p)$ has the highest rate of increase with respect to the variable p. For instance, the maximum value that $\mathcal{F}_{RD}(c,p)$ can attain on the interval $[0, 0.9]$ is less than 100 units, whereas the maximum value of the $\mathcal{F}_P(c,p)$ on the same interval is about 450 units. On the other hand, the penalty function $\mathcal{F}_{DT}(c,p)$ reaches the value 450 units even in a smaller interval $[0, 0.7]$, and grows exponentially on the interval $[0.7, 0.9]$. As a consequence, we claim that the penalty functions $\mathcal{F}_{RD}(c,p), \mathcal{F}_P(c,p)$ and $\mathcal{F}_{DT}(c,p)$ can be approximated by the penalty function of the form $\mathcal{F}_k(c,p)$ for some positive integer value of k.

Thus, we have the following proposition,

Proposition 5.3.1 *: Let $c = 5$ be given, then*

(i) the penalty function $\mathcal{F}_{RD}(c,p)$ can be approximated by $\mathcal{F}_2(c,p)$ on the interval $p \in [0, 9]$,

(ii) The penalty function $\mathcal{F}_P(c,p)$ can be approximated $\mathcal{F}_3(c,p)$ on the interval $p \in [0, 9]$,

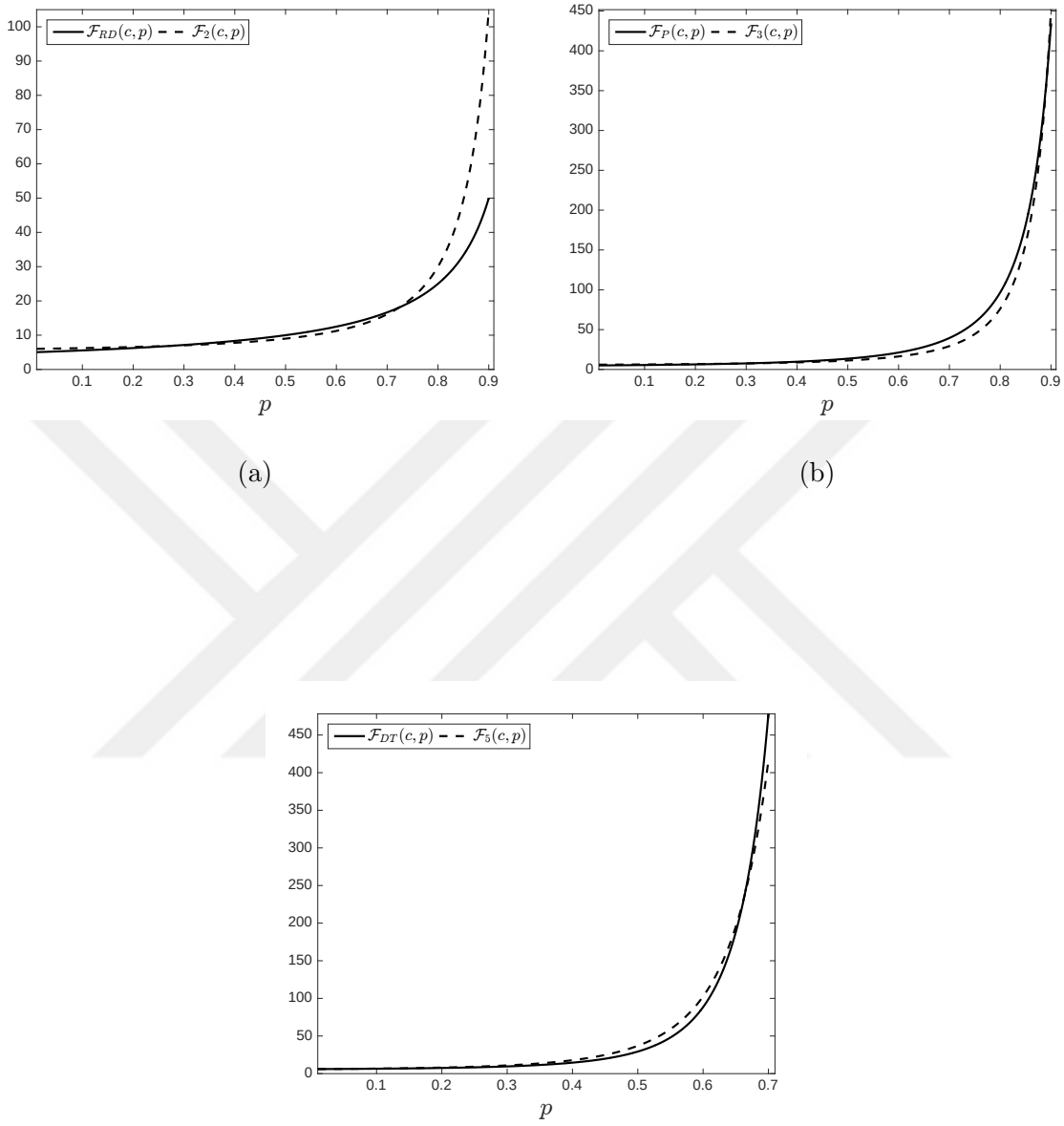


Figure 5.1: Pairwise comparison of the penalty functions $\mathcal{F}_{RD}$, $\mathcal{F}_P$, $\mathcal{F}_{DT}$ with the penalty function $\mathcal{F}_k$ for $k = 2, 3, 5$. Note that both vertical and horizontal values are differently scaled.

(iii) The penalty function $\mathcal{F}_{DT}(c,p)$ can be approximated $\mathcal{F}_5(c,p)$ on the interval $p \in [0,7]$.

Proof : Recall that for any fixed k, the L^2 norm of the function $\mathcal{F}_k(c,p)$ with respect to the variable p in an interval $[a,b]$ is defined as

$$\|\mathcal{F}_k(c,p)\|_{L^2} = \left(\int_a^b |\mathcal{F}_k(c,p)|^2\, dp\right)^{1/2}.$$

Since $\mathcal{F}_k(c,p) < \mathcal{F}_{k'}(c,p)$ for each $p \in [a,b]$ whenever $k < k'$, it is enough to check the following inequalities

$$\begin{aligned}
\|\mathcal{F}_{RD}(c,p) - \mathcal{F}_2(c,p)\|_{L^2} &\leq \|\mathcal{F}_{RD}(c,p) - \mathcal{F}_k(c,p)\|_{L^2} \text{ for } k = 1,3 \text{ on } [0,0.9] \\
\|\mathcal{F}_P(c,p) - \mathcal{F}_3(c,p)\|_{L^2} &\leq \|\mathcal{F}_P(c,p) - \mathcal{F}_k(c,p)\|_{L^2} \text{ for } k = 2,4 \text{ on } [0,0.9] \\
\|\mathcal{F}_{DT}(c,p) - \mathcal{F}_5(c,p)\|_{L^2} &\leq \|\mathcal{F}_{DT}(c,p) - \mathcal{F}_k(c,p)\|_{L^2} \text{ for } k = 4,6 \text{ on } [0,0.7]
\end{aligned}$$

Observe that,

$$\begin{aligned}
\|\mathcal{F}_{RD}(c,p) - \mathcal{F}_0(c,p)\|_{L^2} = 10.9,\ \|\mathcal{F}_{RD}(c,p) - \mathcal{F}_1(c,p)\|_{L^2} = 8.6 \\
\|\mathcal{F}_{RD}(c,p) - \mathcal{F}_2(c,p)\|_{L^2} = 7.7,\ \|\mathcal{F}_{RD}(c,p) - \mathcal{F}_3(c,p)\|_{L^2} = 131.5 \\
\|\mathcal{F}_{RD}(c,p) - \mathcal{F}_3(c,p)\|_{L^2} = 1186.2
\end{aligned}$$

on the interval $[0,0.9]$. Similar argument applies to other cases. Thus the proof follows. □

Corollary 5.3.1.1 *For any fixed value of c there always exist positive integers k_1, k_2, k_3 such that the penalty functions $\mathcal{F}_{RD}(c,p), \mathcal{F}_P(c,p)$, and $\mathcal{F}_{DT}(c,p)$ can be approximated (with respect to the L^2 norm) by the penalty functions of the form $\mathcal{F}_{k_1}(c,p)$, $\mathcal{F}_{k_2}(c,p)$, $\mathcal{F}_{k_3}(c,p)$ respectively on the unit interval $p \in (0,1)$.*

Indeed, we can replace a positive integer value k by a positive real number and find the best k such that the value $\mathcal{F}_{RD}(c,p)$, for instance, is approximated by the value $\mathcal{F}_k(c,p)$. For fixed c, we can find $k := k(c)$ such that $\|\mathcal{F}_{DT}(c,p) - \mathcal{F}_k(c,p)\|_{L^2}$ is minimized on the interval $p \in (0,1)$. Formally,

$$k(c) = \arg\min_k \|\mathcal{F}_{RD}(c,p) - \mathcal{F}_k(c,p)\|_{L^2}$$

As the c value changes the $k(c)$ value also changes, and from Figure 5.2 we observe that there is an upper bound for $k(c)$, i.e, $0 \leq k(c) \leq 2$. Recall that when $c = 5$ and k is a positive integer, we have already proved that for the penalty function $\mathcal{F}_{RD}$ the value of $k(c)$ equals to 2 (see Proposition 5.3.1(i)) which can be also observed from Figure 5.2 as well.

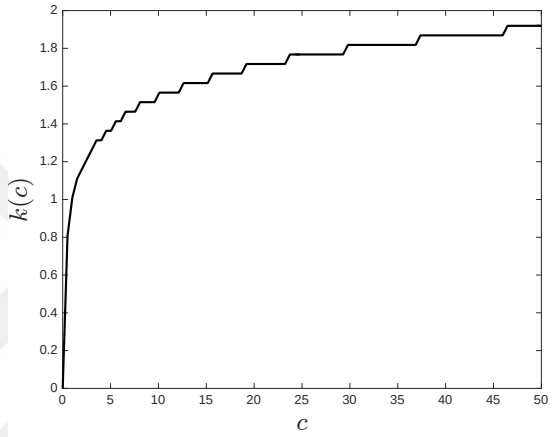


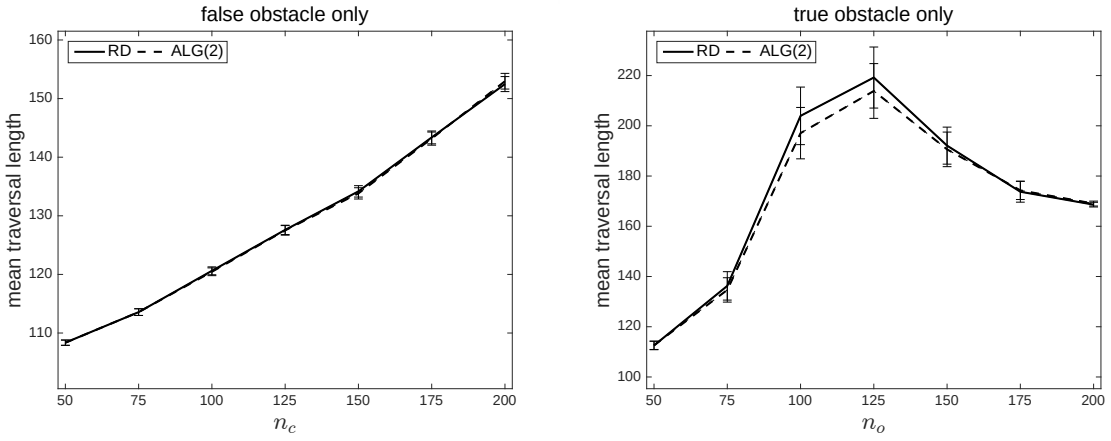
Figure 5.2: Disambiguation cost c versus $k(c)$ values when $p \in (0, 1)$ (when the penalty function $\mathcal{F}_{RD}$ approximated by the penalty function $\mathcal{F}_k$).

5.4 Experimental Setting

Let $i_{max} = j_{max} = 100$ (see Section 1.3) and define the corresponding graph by $G = (V, E)$. We take a starting point s on the upper boundary of the study window, say $s = (50, 100)$. And, for a target point we take a point on the opposite side of square window, say $t = (50, 1)$. These choices are reasonable because, as an example, if we consider the working window as a part of a seashore, then s represents the place where opposing forces (attack position) stand and t represents the place where defensive forces (defense position) protect. Take the radius of obstacles to be $r = 4.5$ units and disambiguation cost $c = 5$ (constant cost). Let p_c and p_o be from distributions F_C=Beta$(4 - \lambda, 4 + \lambda)$ and F_O=Beta$(4 + \lambda, 4 - \lambda)$ where $\lambda = 2$. The

centers $x \in X = X_C \cup X_O$ of disk shaped obstacles are uniformly sampled from $[10, 90] \times [10, 90]$ to ensure that there always exists a (possibly very long) s, t walk. The number of clutter disks we considered are $n_c = 25, 50, 75, 100, 125$ and the number of true obstacles we considered are $n_o = 25, 50, 75, 100, 125$. Finally, to control the precision of the sensor of NAVA assigning probabilities to obstacles, we simulate on various values of λ. If $\lambda = 0$, then sensor has no precision (i.e., NAVA determines its path by a blind sensor). That is to say, the probability of being true obstacle is $1/2$ i.e., NAVA decides by a coin flip about the status of the obstacle. And, if $\lambda = 4$, then sensor has full precision (i.e., NAVA determines its path in perfect detection of true obstacles).

In our simulation setting, we take $\lambda = 2$. These choices are made in line with those of Aksakalli and Ari (2014), Aksakalli and Ceyhan (2012), Ye et al. (2011), and Priebe et al. (2005).



(a) (b)

Figure 5.3: Comparison of mean traversal lengths computed by the RD and ALG(2) algorithms, (a) when there are only clutter disks with $n_c \in \{50, 75, \ldots, 200\}$, (b) and when there are only true obstacles with $n_c = \in \{50, 75, \ldots, 200\}$ with 95% confidence intervals for the mean traversal length computed by the RD and ALG(2) algorithms, respectively.

Let the algorithm using the penalty function $\mathcal{F}_k(c,p)$ be denoted as the ALG(k) algorithm. Based on the plots in Figure 5.1, we claim that the algorithms RD, P, and DT can be characterized by the ALG(k) algorithms for appropriate values of k. Indeed, these three algorithms correspond to ALG(2), ALG(3) and ALG(5), respectively. To support this argument, we computed the traversal length by using these algorithms with 100 Monte Carlo replications. When there are false obstacles only (i.e., $X = X_C$), then it can be seen from Figure 5.3(a) that on the average the RD and ALG(2) algorithms behave almost the same. Similarly, when there are true obstacles only (i.e., $X = X_O$), then it can be see from Figure 5.3(b) that on the average the RD and ALG(2) algorithms behave almost identical where we presented respective 95% confidence intervals for the mean traversal length computed by the RD algorithm and the ALG(2) algorithm.

Moreover, in the mixture case, we show that the mean traversal lengths for different combinations of (n_c, n_o) (see Table 5.1) computed by the RD and ALG(2) algorithms. We observe that the mean traversal length computed by the RD algorithm is well approximated by the mean traversal length computed by the ALG(2) with percentage error at most 2%. The percentage error is referred to the difference between an experimental value and the known value. Formally, if $L_{RD}, L_{ALG(2)}$, are the traversal lengths estimated by the RD and ALG(2) algorithms, respectively. Then the percentage error with respect to L_{RD} is defined as,

$$percentage\ error = \left| \frac{L_{ALG(2)} - L_{RD}}{L_{RD}} \right| \times 100$$

Also, observe that the mean traversal length tends to have larger value when there are more true obstacles than false obstacles. But, when the number of both false and true obstacles are large enough, the study window is saturated by obstacles so that the NAVA circumnavigates and performs no disambiguation (also see Figure 5.3).

Analogously, when NAVA uses the P and ALG(3) algorithms under the same experimental setting, then the empirical comparison of mean traversal lengths are presented in Figure 5.4 and in Table 5.2. We observe that the mean traversal length computed by using the algorithm P is well approximated by using the ALG(3) algo-

		n_o			
		25	50	75	100
n_c	25	**109.41**; 109.48	**119.19**; 119.75	**149.91**; 149.88	**213.71**; 209.02
	50	**115.87**; 116.06	**127.71**; 129.23	**161.41**; 160.36	**184.51**; 183.16
	75	**120.95**; 121.23	**134.99**; 135.37	**164.68**; 166.34	**175.61**; 174.52
	100	**128.75**; 128.77	**145.98**; 145.45	**177.11**; 175.88	**169.16**; 170.12

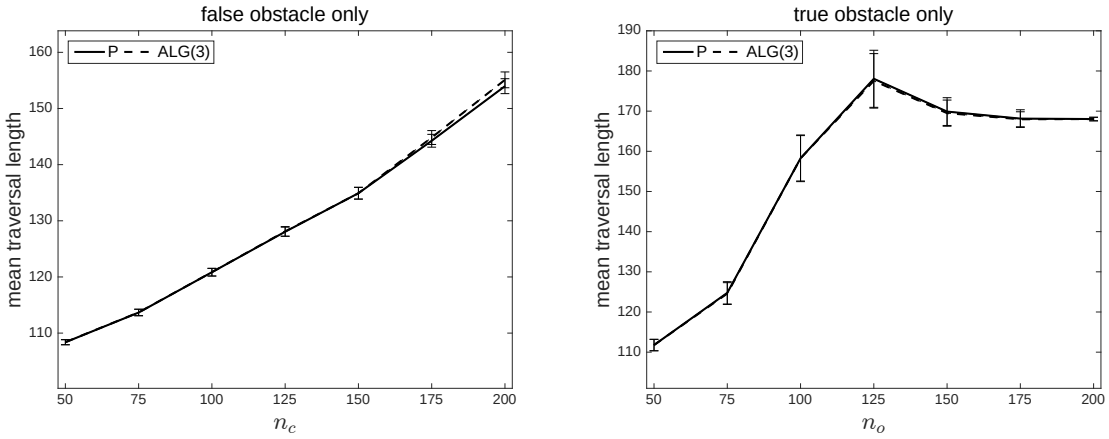
Table 5.1: Comparison of mean traversal lengths estimated by the RD (shown in bold) and ALG(2) algorithms for various pairs of (n_c, n_o) with percentage error less than 2%.

rithm. Moreover, comparing the performances of the RD and P algorithms (hence ALG(2) and ALG(3)), on the average the mean traversal length computed by the P algorithm is less than the mean traversal length computed by the RD algorithm i.e., specifically when there are more true obstacles than false obstacles (Table 5.1 versus Table 5.2). Analogous trends can be observed for the mean traversal length

		n_o			
		25	50	75	100
n_c	25	**109.50**; 109.54	**117.68**; 117.63	**139.16**; 139.19	**175.87**; 176.63
	50	**115.24**; 115.30	**125.66**; 125.62	**146.16**; 146.40	**166.08**; 166.93
	75	**121.11**; 121.16	**133.27**; 133.52	**152.65**; 152.69	**164.23**; 163.33
	100	**128.94**; 128.93	**143.30**; 143.55	**161.17**; 161.96	**166.28**; 166.21

Table 5.2: Comparison of mean traversal lengths computed by the P (shown in bold) and ALG(3) algorithms for various pairs of (n_c, n_o) with percentage error less than 0.5%

computed by the DT and ALG(5) algorithms (see Figure 5.5 and Table 5.3). But, when comparing the empirical performances of the DT and P algorithms, it turns out



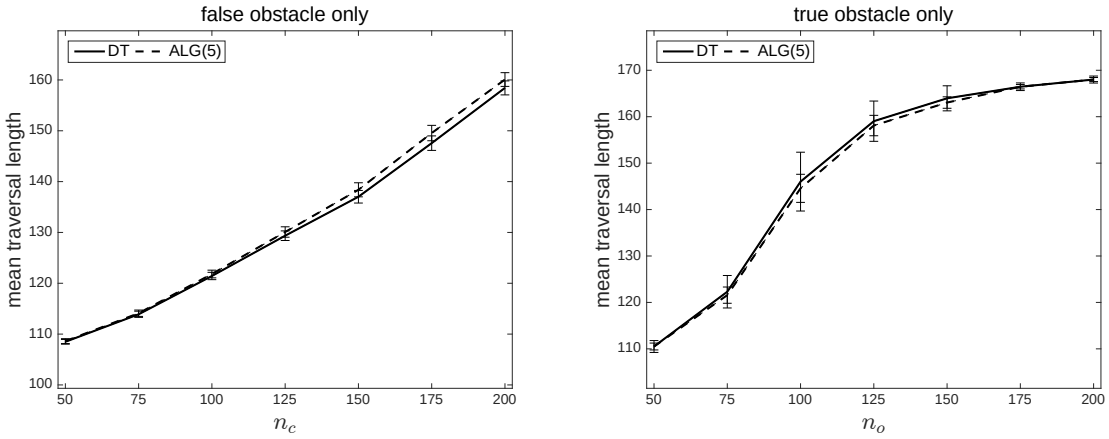
(a) (b)

Figure 5.4: Comparison of mean traversal lengths estimated by the P and ALG(3) algorithms, (a) when there are only clutter disks with $n_c \in \{50, 75, \ldots, 200\}$, (b) and when there are only true obstacles with $n_c = \in \{50, 75, \ldots, 200\}$ with 95% confidence intervals for the mean traversal length computed by the P and ALG(3) algorithms, respectively.

that the mean traversal length computed by the DT algorithm tends to have smaller values than the mean traversal length computed by the P algorithm (Table 5.2 versus Table 5.3).

Thus, we empirically observed that when the disambiguation cost $c = 5$ together with the sensor precision variable $\lambda = 2$, the algorithms RD, P, and DT can be closely approximated by the ALG(2), ALG(3) and ALG(5) algorithms. This result can be extended to the general case, but this time when approximating the RD, P, and DT algorithms note that the k value in the ALG(k) algorithm may change as the disambiguation cost c changes (see Figure 5.2).

Proposition 5.4.1 *: Let $c = 5$, then for any fixed λ the mean traversal lengths of NAVA computed by the algorithms RD, P and DT can be approximated by the mean traversal lengths computed by the ALG(k) algorithms for $k = 2, 3, 5$, respectively.*



(a) (b)

Figure 5.5: Comparison of mean traversal lengths estimated by the DT and ALG(5) algorithms, (a) when there are only clutter disks with $n_c \in \{50, 75, \ldots, 200\}$, (b) and when there are only true obstacles with $n_o \in \{50, 75, \ldots, 200\}$ with 95% confidence intervals for the mean traversal length computed by the DT and ALG(5) algorithms, respectively.

		n_o			
		25	50	75	100
n_c	25	**109.74**; 109.63	**116.94**; 117.14	**135.99**; 133.01	**153.03**; 152.02
	50	**116.48**; 116.83	**126.57**; 127.24	**145.12**; 145.09	**157.59**; 157.32
	75	**121.54**; 122.05	**135.71**; 136.48	**155.54**; 155.52	**164.68**; 163.98
	100	**129.94**; 130.34	**144.52**; 147.09	**161.31**; 162.00	**167.19**; 167.17

Table 5.3: Comparison of mean traversal lengths computed by the DT (shown in bold) and ALG(5) algorithms for various pairs of (n_c, n_o) with percentage error less than 1%.

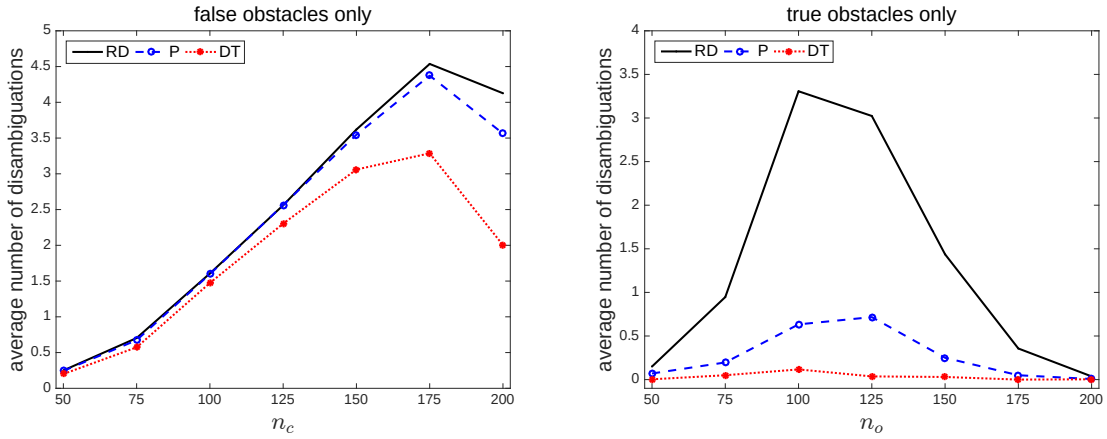
Proof : The proof follows from the proposition 5.3.1. □

In the false obstacle only case, the traversal length computed by all three RD,

P, and DT algorithms (hence ALG(2), ALG(3), and ALG(5)) give similar results (Figures 5.3(a), 5.4(a), and 5.5(a)), but in the true obstacle only case the worst performing one is the RD algorithm and the best performing one is the DT algorithm (Figures 5.3(b), 5.4(b), and 5.5(b)). The DT algorithm avoids disambiguations when encountered by true obstacles on its way, whereas both the RD and P algorithms do take risk and do not attempt to choose the immediate zero-risk path (unlike the DT algorithm). This situation can be explained by analyzing the total number of disambiguations in computing the traversal length by all these algorithms. In Figure 5.6(a), the false obstacle only case, the average number of disambiguations carried out by the RD and P algorithms are almost same, whereas it decreases with the DT algorithm as the number of false obstacles increase. On the other hand, in Figure 5.6(b), the true obstacle only case, the average number of disambiguations performed by the RD algorithm increases first, reaches its peak, and decreases as the number of true obstacles increase. As for the P algorithm, the average number of disambiguations is always less than 1. Interestingly, the average number of disambiguations used by the DT algorithm is close to zero, i.e., the DT algorithm avoids disambiguations when encountered by the true obstacles on its way, takes no risk, and inclined to choose the plain zero-risk path. So far, we have investigated the trends in mean traversal length computed by the ALG(k) algorithms for $k = 2, 3, 5$ when the disambiguation cost $c = 2$, and the sensor precision parameter $\lambda = 2$. The next natural question would be how exactly the ALG(k) algorithm depends on sensor's detection parameter λ if the value of c is fixed.

5.5 ALG(k) Algorithm and Sensor Precision

Prior to the traversal of NAVA, the corresponding marks (probabilities of non-traversability) of disk shaped obstacles are assigned by the sensor of NAVA. The p_c and p_o come from distribution functions F_C and F_O with Beta$(4-\lambda, 4+\lambda)$ and Beta$(4+\lambda, 4-\lambda)$ distributions. The higher probability values indicate higher probability of non-traversability, and vice versa. If $\lambda = 0$, then NAVA has no capability of



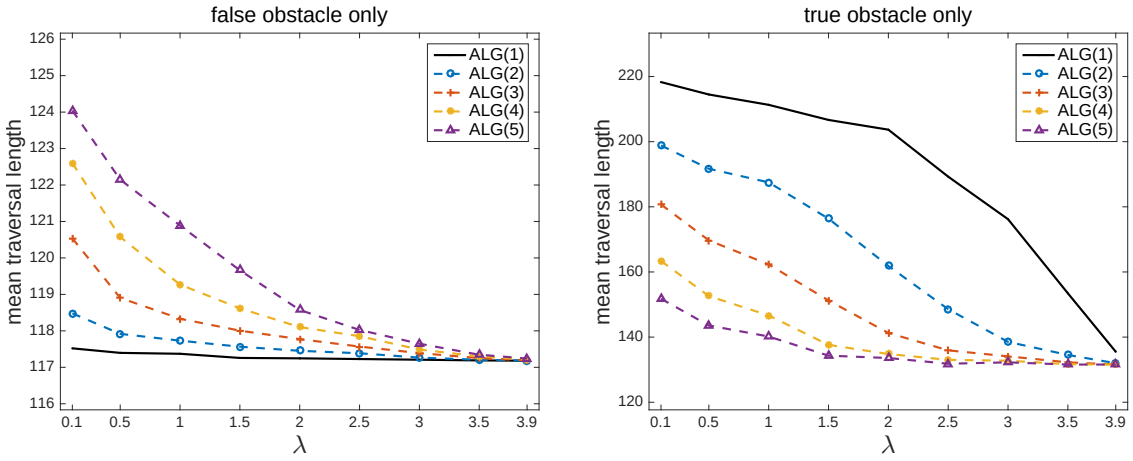
(a) (b)

Figure 5.6: Comparison of average number of disambiguations when the mean traversal length is estimated by the RD, P, and DT algorithms, (a) when there are only clutter disks with $n_c \in \{50, 75, \ldots, 200\}$, (b) and when there are only true obstacles with $n_o \in \{50, 75, \ldots, 200\}$. Note that vertical axes are differently scaled.

distinguishing true obstacles from clutter (distinguishes them in random with probability 1/2). If $\lambda = 4$, then NAVA has perfect sensing capability to detect true and false obstacles.

In Figure 5.7(a) as the λ value increases (i.e., sensor precision increases), on the average the mean traversal length decreases and converges to some value (i.e., the benchmark value) with each ALG(k) algorithm for $k = 1, 2, 3, 4, 5$, but at different rates of convergence. When there are only clutter disks, on the average the ALG(1) algorithm performs best and the ALG(5) is the worst one. Similarly, in Figure 5.7(b), the mean traversal length again decreases and converges to some value (i.e., the benchmark value) under each algorithm ALG(k) for $k = 1, 2, 3, 4, 5$, but at different rates of convergence. When there are only true obstacles, now the reverse ordering is observed. In this case, on the average the ALG(5) algorithm is the best and the ALG(1) is the worst one. Hence, the following proposition holds.

Proposition 5.5.1 *: Let $L_{ALG(k)}(\lambda)$ be the traversal length computed by the algorithm*



(a) (b)

Figure 5.7: Comparison of mean traversal lengths estimated by the ALG(k) algorithms, (a) when there are only clutter disks with $n_c = 50, 75, 100, 125$, (b) and when there are only true obstacles with $n_c = 50, 75, 100, 125$. Notice that vertical axes are differently scaled. Note that vertical axes are differently scaled.

ALG(k) under the sensor precision parameter λ for the given problem setting (see Section 5.4). Then,

- *For fixed k and for any $\lambda > \lambda'$ we have $L_{ALG(k)}(\lambda) \leq_{st} L_{ALG(k)}(\lambda')$ when $X = X_C$ or $X = X_O$,*
- *For fixed λ and for any $k > k'$ we have $L_{ALG(k)}(\lambda) \geq_{st} L_{ALG(k')}(\lambda)$ when $X = X_C$,*
- *For fixed λ and for any $k > k'$ we have $L_{ALG(k)}(\lambda) \leq_{st} L_{ALG(k')}(\lambda)$ when $X = X_O$,*

where '$\geq_{st}$' stands for 'stochastically larger than' and '$\leq_{st}$' stands for 'stochastically smaller than'.

Proof : (Sketch) (a) Let k be given and suppose that $X = X_C$ (clutter only case). For

any disk shaped false obstacle D_0 let p_λ be its probability of being true obstacle coming from the distribution Beta$(4-\lambda, 4+\lambda)$. Next, suppose that we change the precision of sensor so that it assigns $p_{\lambda'}$ for D_0 coming from the distribution Beta$(4-\lambda', 4+\lambda')$. Since the corresponding means of distributions Beta$(4-\lambda, 4+\lambda)$ and Beta$(4-\lambda', 4+\lambda')$ are $(4-\lambda)/8$ and $(4-\lambda')/8$, then for $\lambda > \lambda'$ we have $p_\lambda \leq_{st} p_{\lambda'}$. That is to say, as λ value increases (increase in precision) the detection of the probability of disk shaped obstacle being a true obstacle increases, and as a result the penalty value coming from $\mathcal{F}_k(c,p)$ increases as well. In this case, it is more likely for NAVA to choose longer path rather than shorter path. Hence, $L_{ALG(k)}(\lambda) \leq_{st} L_{ALG(k)}(\lambda')$. Similar arguments apply for the case $X = X_O$ (true obstacle only case).

(b) For any fixed λ (sensor precision is fixed) as k value increases the penalty function $\mathcal{F}_k(c,p)$ also increases. As penalty value increases it is more likely for NAVA to choose a longer path. Hence, $L_{ALG(k)}(\lambda) \geq_{st} L_{ALG(k')}(\lambda)$ whenever $k > k'$ and $X = X_C$ (clutter only case).

(c) The idea is similar to (b). When there are true obstacles only ($X = X_O$), then small values of k lead to increase the number of disambiguations and the traversal length. However, for large values of k the penalty values increase so as to avoid potential disambiguations and hitting true obstacles. Thus, for $k > k'$ we have the relation $L_{ALG(k)}(\lambda) \leq_{st} L_{ALG(k')}(\lambda)$.

In both Figures 5.7(a)-(b), the mean traversal length monotonically decreases as the λ approaches to 4. When the λ is sufficiently close to 4, the sensor precision of NAVA becomes perfect in detecting the true obstacles. That is to say, $p_c \approx 0$ and $p_o \approx 1$, and recall that in computing the benchmark value the corresponding probabilities p_c and p_o were set to be 0 and 1 because NAVA knows the status of all obstacles prior to traversal. Hence,

Corollary 5.5.1.1 *In false or true obstacle only cases, the mean traversal length estimated by the algorithm ALG(k) converges to the mean traversal length estimated by the benchmark algorithm as $\lambda \to 4$ for any fixed value of k, i.e.,*

$$L_{ALG(k)}(\lambda) \to L_B \text{ as } \lambda \to 4$$

whenever $X = X_C$ *or* $X = X_O$.

From Proposition 5.5.1, we can choose the best performing algorithm when λ is given and $X = X_C$ or $X = X_O$. We still do not know how to choose the best performing algorithm when there is a mixture of both false and true obstacles uniformly generated in the traversal environment.

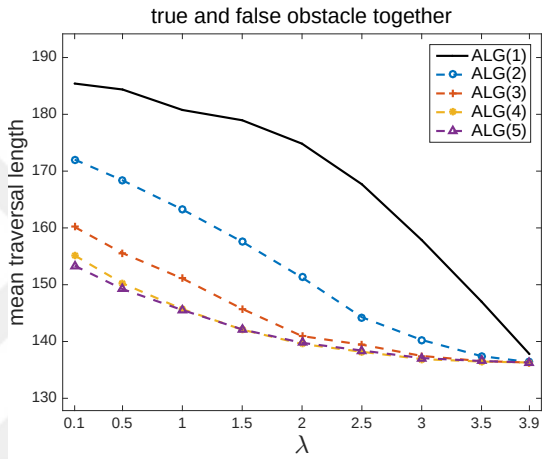


Figure 5.8: Comparison of mean traversal lengths estimated by the ALG(k) algorithms when there are total $n_c + n_o$ number of obstacles uniformly generated with $n_c = 25, 50, 75, 100$ and $n_o = 25, 50, 75, 100$.

In Figure 5.8, there are a total of $n_c + n_o$ obstacles uniformly generated with $n_c = 25, 50, 75, 100$ and $n_o = 25, 50, 75, 100$. Apparently, ALG(5) is the overall best algorithm. We already observed that the preference of ALG(k) may change according to the number of false and true obstacles. To illustrate this, let us investigate the case when $n_c + n_o = 100$. Because of our simulation setting, this can only occur with the pairs $(n_c, n_o) = (25, 75), (50, 50), (75, 25)$. In Figure 5.9(a), as expected ALG(5) is the best choice to find the shortest route for the case $(n_c, n_o) = (25, 75)$, since, ALG(5) performs better than others when there are more true obstacles than false obstacles (see Figure 5.7(b)). In Figure 5.9(b), we observe that both ALG(4) and ALG(5) perform better than others with the case $(n_c, n_o) = (50, 50)$. However, in

Figure 5.9(c) ALG(2) and ALG(3) are better than others. Thus, in the uniform case the choice of the algorithm depends on the number of true obstacles n_o, on the number clutter disks n_c, and the sensor parameter λ. The only common thing in Figures 5.8

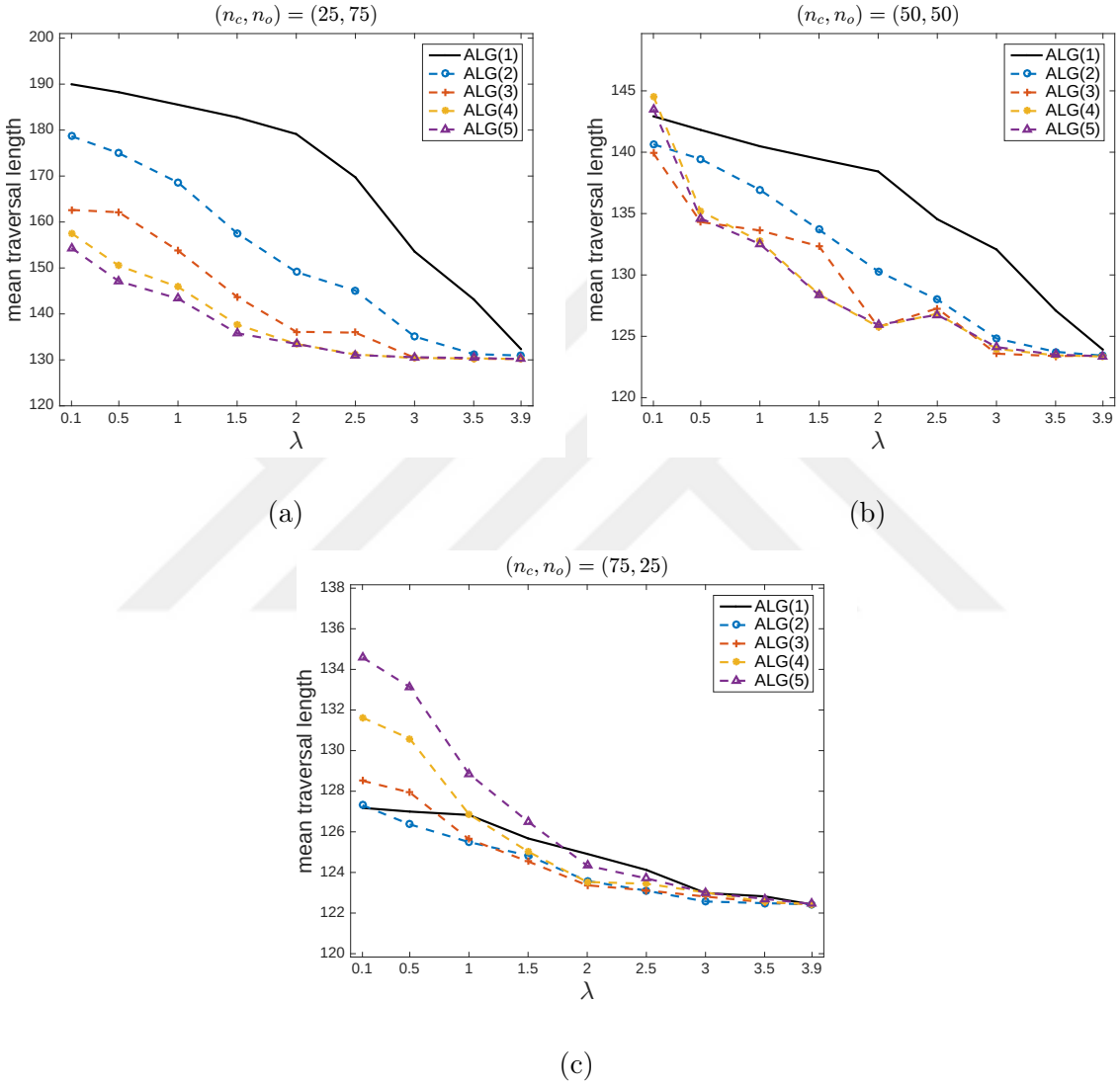


Figure 5.9: Comparison of mean traversal lengths estimated by the ALG(k) algorithms, (a) for $(n_c, n_o) = (25, 75)$, (b) for $(n_c, n_o) = (50, 50)$, (c) and for $(n_c, n_o) = (75, 25)$. Notice that vertical axes are differently scaled.

and 5.9 is that the mean traversal length converges to the benchmark value as the sensor precision of NAVA increases. Thus, we can generalize the Corollary 5.5.1.1

Corollary 5.5.1.2 *The mean traversal length estimated by the ALG(k) algorithm converges to the mean traversal length estimated by the benchmark algorithm as* $\lambda \to 4$ *for any fixed value of* k*, i.e.,*

$$L_{ALG(k)}(\lambda) \to L_B \text{ as } \lambda \to 4$$

whenever $X = X_C \cup X_O$*.*

In the uniform case, we claim that when the triple (n_o, n_c, c, λ) is given, then we can choose the best performing algorithm ALG(k) by modifying the penalty function $\mathcal{F}_k(c, p)$ where $k := k(n_o, n_c, c, \lambda)$. Another approach would be considering the ratio $\rho = n_o/n_c$. For our special cases of experimental setting, if $\rho \leq 1$, then we recommend choosing the ALG(2) or ALG(3) algorithm. If $\rho > 1$, then choose the ALG(4) or ALG(5) algorithm. For the extreme cases, if $n_o = 0$ then it is reasonable to choose ALG(1). And, if $n_c = 0$ then it is reasonable to choose the ALG(5) algorithm.

5.6 ALG(k) Algorithm and the Disambiguation Cost

In Section 5.5, we have discussed about the behaviour of the ALG(k) algorithm when the disambiguation cost was taken to be $c = 5$. But, when the constant disambiguation cost changes then a question of interest is which type of algorithm should be chosen among the ALG(k)'s. We already investigated the trends in mean traversal length when the sensor parameter λ changes. So, to keep it simple we will fix $\lambda = 2$ and consider the disambiguation cost $c = 1, 3, 5, 7, 9$ units. The centers of disk shaped obstacles are again uniformly generated inside the working window.

In Figure 5.10, we observe that the ALG(3), ALG(4), and ALG(5) algorithms clearly outperform the others. But, as we have seen in Figure 5.8 this might not be the general case. For a detailed investigation, consider the case when $n_c + n_o = 100$ which occurs for the pairs $(n_c, n_o) = (25, 75), (50, 50), (75, 25)$.

In Figures 5.11(a-b), we observe that the ALG(5) algorithm outperforms the others. Indeed, the performance of the ALG(4) algorithm is similar to the performance of the ALG(5) algorithm with a slight difference. But, in Figure 5.11(c) we have

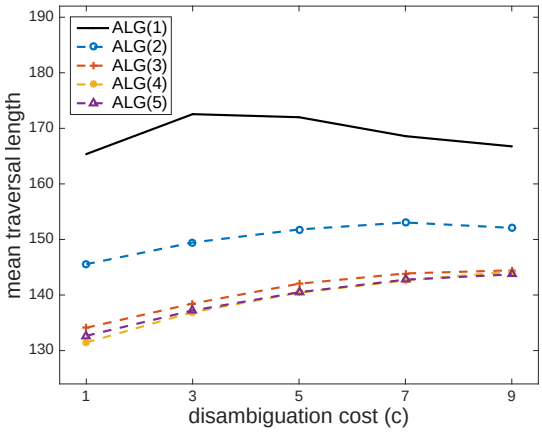


Figure 5.10: Comparison of mean traversal lengths estimated by the ALG(k) algorithms when there are total $n_c + n_o$ number of obstacles (combination of all (n_c, n_o) pairs) uniformly generated with $n_c, n_o \in \{25, 50, 75, 100\}$.

different candidates for the best performing algorithm i.e., the ALG(2) and ALG(3). And, even in some cases the algorithm ALG(1) gives smaller mean traversal length than others. These results are consistent with those in Section 5.5. Also, observe that as the disambiguation cost value c increases, the mean traversal length increases as well, and for large values of c the mean traversal length tends to be equal to the zero-risk path. Since, as c value increases disambiguating obstacles gets expensive for all algorithms, ALG(k) algorithms will eventually avoid disambiguations whenever encounter true obstacles on the way of NAVA's traversal.

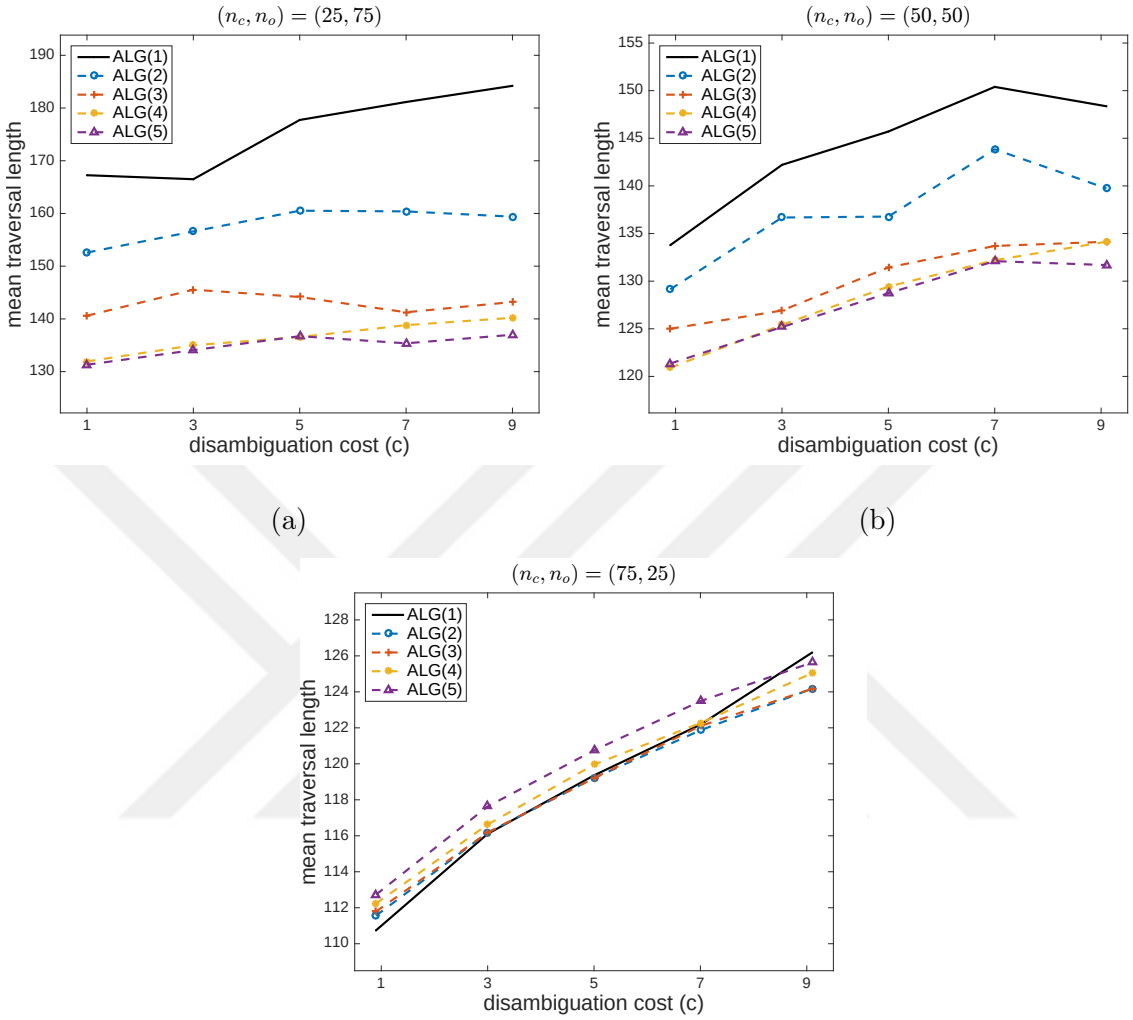
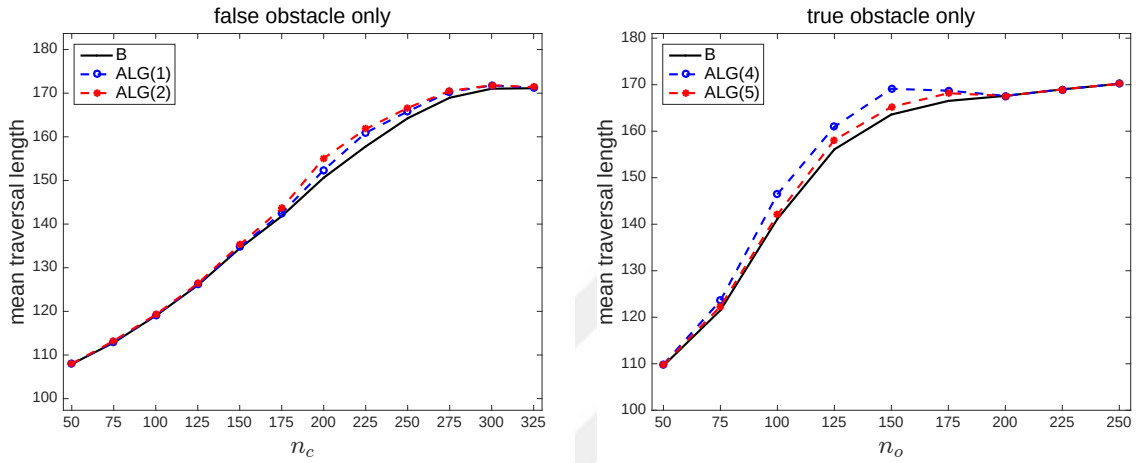


Figure 5.11: Comparison of mean traversal lengths estimated by the ALG(k) algorithms, (a) for $(n_c, n_o) = (25, 75)$, (b) for $(n_c, n_o) = (50, 50)$, (c) and for $(n_c, n_o) = (75, 25)$. Notice that vertical axes are differently scaled.

5.7 Convergence of Traversal Length Under ALG(k) Algorithm

In this section, we investigate the behavior of the ALG(k) algorithm for nonnegative values of k as the number of obstacles increases. Suppose that, there are only clutter disks in the working environment. Then, previous sections suggest using the ALG(1)

algorithm to attain the shortest s, t path. Also, if there are only true obstacles, then the better performing algorithms are the ALG(k)'s with large k values. For simplicity, in our experimental computation we fix $\lambda = 2, c = 5$.



(a) (b)

Figure 5.12: (a) Mean traversal length estimated by benchmark, ALG(1), ALG(2) algorithms versus n_c with $n_c \in \{50, 75, \ldots, 325\}$. (b) Mean traversal length estimated by the benchmark, ALG(4), ALG(5) algorithms versus n_o with $n_o \in \{50, 75, \ldots, 250\}$. Notice that horizontal axes are differently scaled.

Figure 5.12 suggests that the best performing algorithm in finding the shortest s, t walk when there are clutter disks only is the ALG(k) algorithm with smaller k and when there are true obstacles only is ALG(k) algorithm with larger k, respectively. Indeed, if $X = X_C$ (false obstacle only) then taking $k = 0$ will give the benchmark value for the mean traversal length. As we have discussed before, the benchmark value is the optimal (i.e., lowest) value that any heuristic algorithm can achieve. If $X = X_O$ (true obstacle only), then taking $k = 5$ or larger will give approximately the benchmark value for the mean traversal length. Thus, we have the following theorem

Theorem 5.7.1 *: For fixed c and λ, let L_B and $L_{ALG(k)}(\lambda)$ be the traversal lengths computed by the benchmark (B) algorithm and the ALG(k) algorithm, respectively.*

Then,

1. $L_{ALG(k)}(\lambda) = L_B$ *for* $k = 0$ *when* $X = X_C$

2. $L_{ALG(k)}(\lambda) \searrow L_B$ *as* k *tends to* ∞ *when* $X = X_O$

Proof : (Sketch) (1) When $X = X_C$, then in computing the benchmark value, NAVA knows location and status of all obstacles. In this case, NAVA does not need to use the sensor, and thus probability of being true obstacle is irrelevant. Discarding the probability is equivalent to setting $k = 0$ in the penalty function $\mathcal{F}_k(c, p)$. Hence, the traversal length computed by the ALG(0) algorithm is same as the the traversal length computed by the B algorithm, and equality $L_{ALG(k)}(\lambda) = L_B$ holds for $k = 0$ with any λ.

(2) For any $k > k'$, if NAVA disambiguates some disk shaped obstacle D_0 using the penalty function $\mathcal{F}_k$, then it will also disambiguate D_0 when using the penalty function $\mathcal{F}_{k'}$ ($\mathcal{F}_k(c, p)$ is increasing with respect to k), and thus will follow the same path. If NAVA does not disambiguate under the ALG(k) algorithm and continues its route using path π_0, but it disambiguates some obstacle D_1 under the ALG(k') algorithm, then NAVA will learn the actual status of an obstacle (i.e., true obstacle) at a constant cost c and will force to replan its route to some path π_1 with length greater than or equal to π_0. If NAVA does not disambiguate under neither the ALG(k) nor ALG(k') algorithm, then computed traversal lengths will be identical. Hence, we obtain a decreasing sequence $\{L_{ALG(k)}(\lambda)\}_{k\in\mathbb{N}}$ converging to L_B. Taking k large enough means appointing very large penalty value to an edge intersecting the obstacle which is equivalent to deleting the blocked edge. So, $L_{ALG(k)}(\lambda) \searrow L_B$ follows as $k \to \infty$.

But, we still do not have an efficient algorithm when there are both clutter and true obstacles in the environment. From the idea of Theorem 5.7.1, we construct a new algorithm called T(α) (T comes from the word threshold), where $\alpha \in (0, 1)$ is the probability threshold in which an obstacle should or should not be disambiguated. When a problem instance is given, the main idea of the T(α) algorithm is choosing the value k in penalty function $\mathcal{F}_k(c, p)$ by using the threshold parameter α. If $\mathcal{F}_{T(\alpha)}$

is the penalty function used by the algorithm T(α), then formally,

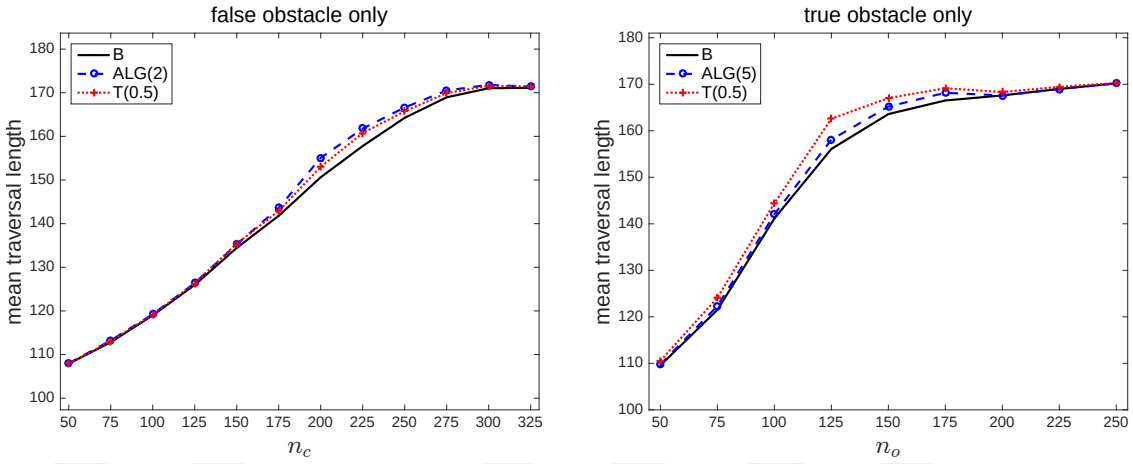
$$\mathcal{F}_{T(\alpha)}(c,p) = \begin{cases} \mathcal{F}_k(c,p) \text{ with } k=0 \text{ if } p < \alpha \\ \mathcal{F}_k(c,p) \text{ with } k = \frac{\log(z_0-c)}{\log(1/(1-\alpha))} \text{ if } p \geq \alpha \end{cases}$$

where z_0 is the traversal length of the zero risk path (computed by the Z algorithm) from s to t.

Since the obstacle centers are sampled uniformly inside the region $[10,90]\times[10,90]$ and $r = 4.5$ then as the number of obstacles increases, the region $[5,95] \times [5,95]$ is extensively occupied. So, to simplify our computation in simulations, it is reasonable to assign the value of the z_0 as the Euclidean distance following the vertices $s, (95,95), (95,5), t$ which yields 180.45 units. For example, when $c = 5$ and threshold $\alpha = 0.5$ we obtain the corresponding $k = 7.5$. NAVA is only given the corresponding probabilities of disks being true obstacles, so choose the threshold $\alpha = 0.5$. Otherwise, loosening (i.e., decreasing α) or strengthening (i.e., increasing α) the threshold value might cause the loss of clutter or true obstacle information which is not a desirable situation, as it might cause instability in the performance of the algorithm.

In Figure 5.13, we observe that the empirical performance of the algorithm T(0.5) is similar to the ALG(2) algorithm and the ALG(5) algorithm when there are only false obstacles and only true obstacles, respectively. So, in general when there are both clutter and true obstacles, on the average the mean traversal length computed by the T(0.5) algorithm does not converge to the benchmark value. The main drawback of the penalty function $\mathcal{F}_{T(0.5)}$ is that it strictly separates clutter and true obstacles from each other by the given threshold $\alpha = 0.5$.

To overcome this drawback, we increase the number of thresholds. Let us divide the interval $[0,1]$ into 20 equal subintervals with length 0.05 units. And, let the vector $I = (0, 0.05, 0.10, \ldots, 0.95, 1.00)$ be the corresponding end points. Similarly, for any fixed positive integer $\tilde{k}$ define the vector $K = (0, \tilde{k}/10, 2\tilde{k}/10, \ldots, 10\tilde{k}/10, \ldots, 20\tilde{k}/10)$. If $p_1, p_2, \ldots, p_{|X|}$ are the probabilities (of being a true obstacle) of disk shaped obstacles, then let m_0 be the mean of $p_i's$, i.e.,



(a) (b)

Figure 5.13: (a) Mean traversal length versus n_c computed by the B, ALG(2), T(0.5) algorithms with $n_c \in \{50, 75, \ldots, 325\}$. (b) Mean traversal length versus n_o computed by the B, ALG(5), T(0.5) algorithms with $n_o \in \{50, 75, \ldots, 250\}$. Note that horizontal axes are differently scaled.

$m_0 = \frac{1}{m'} \sum_{i=0}^{|X|} p_i$. Then, perform the following operation

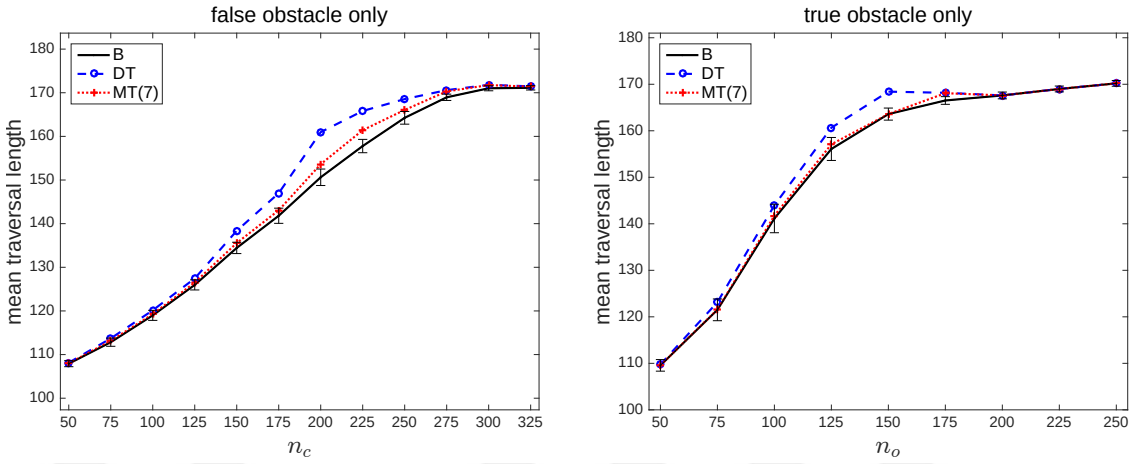
$$\text{Set } K(i) = K(i) + (i-11)\tilde{k}/10 \quad \text{if} \quad I(i) \leq m_0 < I(i+1) \text{ for } i = 1, 2, ..., 10,$$

$$\text{and, } K(i) = K(i) + (i-10)\tilde{k}/10 \quad \text{if} \quad I(i) \leq m_0 < I(i+1) \text{ for } i = 11, 12, ..., 20.$$

At the end of this operation, some values of vector K might be negative depending on m_0. To avoid this, set $K(i) = 0$ if $K(i)$ is negative and $K(i) = 2\tilde{k}$ if $K(i)$ is greater than $2\tilde{k}$. Now, consider the following penalty function,

$$\mathcal{F}_{MT(\tilde{k})}(c, p) = \mathcal{F}_k(c, p) \text{ with } k = K(i) \quad \text{if} \quad I(i) \leq p < I(i+1) \text{ for } i = 1, 2, ..., 20.$$

In Figure 5.14, we observe that the new MT($\tilde{k}$) algorithm (MT comes from words mean and threshold) with $\tilde{k} = 7$ is better than the DT algorithm when there are either only false obstacles or only true obstacles. Indeed, for these special cases on the average the benchmark value is well estimated by the traversal length estimated by the $\mathcal{F}_{MT(7)}(c, p)$ penalty function.

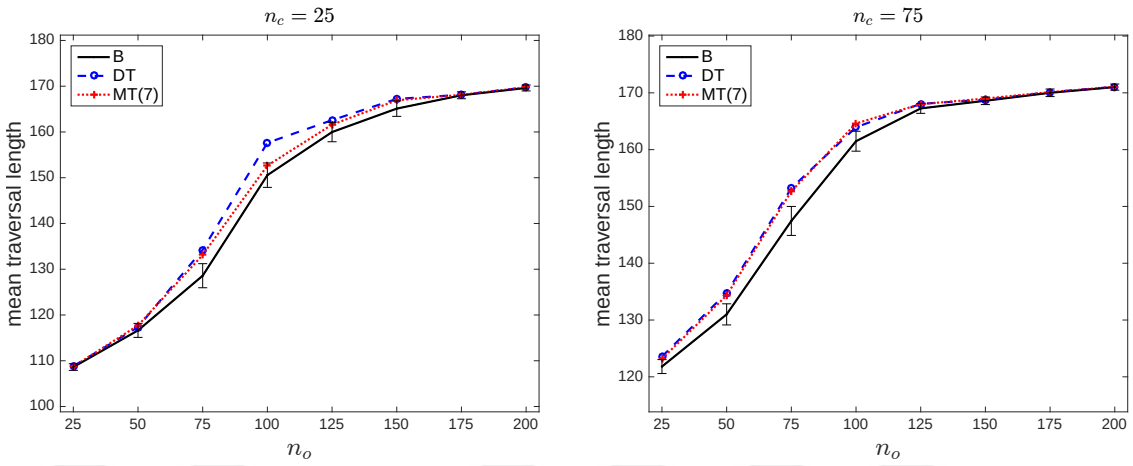


(a) (b)

Figure 5.14: (a) Mean traversal length versus n_c estimated by the B, DT, MT(7) algorithms with $n_c \in \{50, 75, \ldots, 325\}$. (b) Mean traversal length versus n_o computed by the B, DT, MT(7) algorithms with $n_o \in \{50, 75, \ldots, 250\}$ with 95% confidence intervals for the benchmark value. Notice that horizontal axes are differently scaled.

To investigate the new algorithm in detail, we consider it for different pairs of (n_c, n_o). Thus, in Figure 5.15 we observe that on the average the MT($\tilde{k}$) algorithm with $\tilde{k} = 7$ outperforms the DT algorithm. For example, in Figure 5.15(a) when $(n_c, n_o) = (25, 100)$ the mean traversal length computed by the algorithm MT(7) is smaller than the mean traversal length computed by the DT algorithm.

The fact that when there are mixture of both true and false obstacles, the MT($\tilde{k}$) algorithm does not converge to the benchmark value, encourages us to improve the MT($\tilde{k}$) algorithm. Instead of dividing the unit interval $[0, 1]$ into 20 equal parts in the construction of the algorithm, we divide the interval into n (default $n = 100$) equal parts. And, we choose $\tilde{k} = 2, 4, 6, 8$ in our setting (see Algorithm 1). Using this algorithm, we estimate the benchmark value for different pairs (n_c, n_o) with 100 Monte Carlo replications. In Figures 5.16 and 5.17, the mean traversal length estimated by the improved MT($\tilde{k}$) algorithm almost coincides with the benchmark value. Deter-



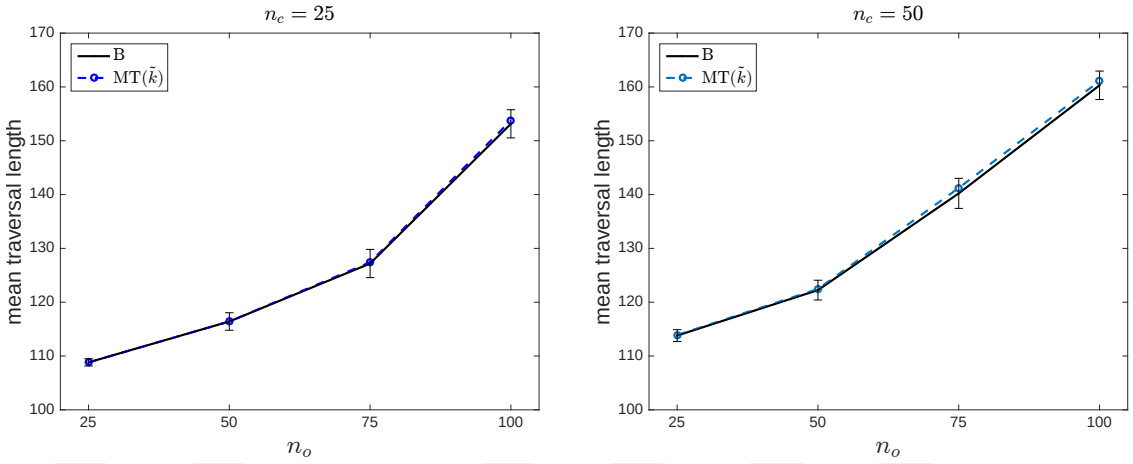
(a) (b)

Figure 5.15: Comparison of mean traversal lengths estimated by the B, DT, MT(7) algorithms, (a) for $n_c = 25$ with $n_o \in \{25, 50, \ldots, 200\}$, (b) and for $n_c = 75$ with $n_o \in \{25, \ldots, 200\}$ with 95% confidence intervals for the benchmark value.

mining the optimal choice of $\tilde{k}$ to approximate the benchmark value more accurately is a topic of ongoing research.

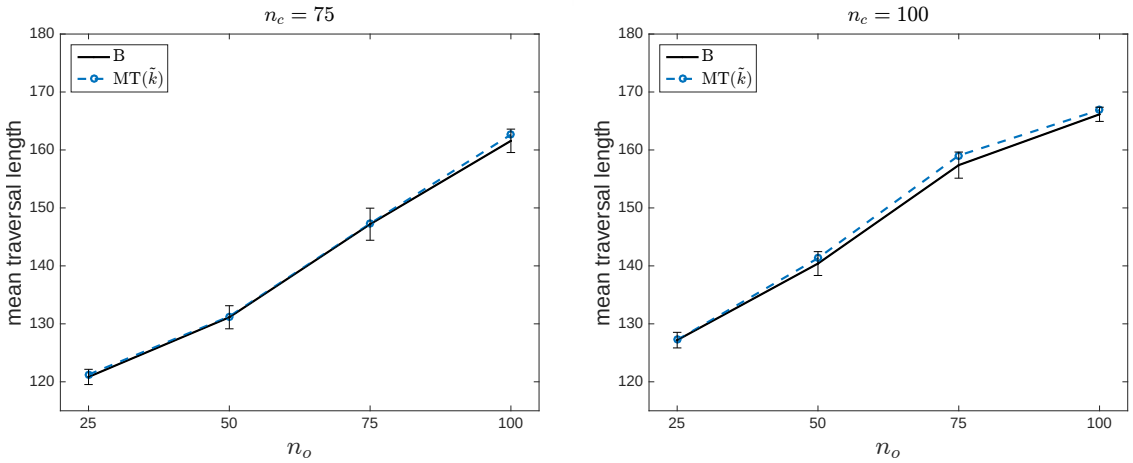
So far, we have discussed the penalty functions $\mathcal{F}_k(c, p)$ and their empirical performances for positive integer k values. We also developed various effective algorithms converging to the benchmark value for some special cases. Although the MT($\tilde{k}$) algorithm with $\tilde{k} = 7$ outperforms the DT algorithm, in general it does not converge to the benchmark value. Moreover, by improving the MT($\tilde{k}$) algorithm we deduced better estimations for the benchmark value by considering $\tilde{k} = 2, 4, 6, 8$.

On the other hand, for any given problem setting, on the average the benchmark value can also be approximated by the ALG(k) algorithm with appropriate choice of positive integer k values. From Figures 5.18 and 5.19 we observe that on the average the benchmark value is well approximated by the ALG(k) algorithm. In our simulations, we considered the integer values of k between 0 and 6 with 100 Monte Carlo replications. For each pair of (n_c, n_o), we used the best possible penalty



(a) (b)

Figure 5.16: Comparison of mean traversal lengths estimated by the B and $MT(\tilde{k})$ algorithms, (a) for $n_c = 25$ with $n_o = 25, 50, 75, 100$, (b) and for $n_c = 50$ with $n_o = 25, 50, 75, 100$ with 95% confidence intervals for the benchmark value.



(a) (b)

Figure 5.17: Comparison of mean traversal lengths estimated by the B and $MT(\tilde{k})$ algorithms, (a) for $n_c = 75$ with $n_o \in \{25, 50, \ldots, 100\}$, (b) and for $n_c = 100$ with $n_o \in \{25, 50, \ldots, 100\}$ with 95% confidence intervals for the benchmark value.

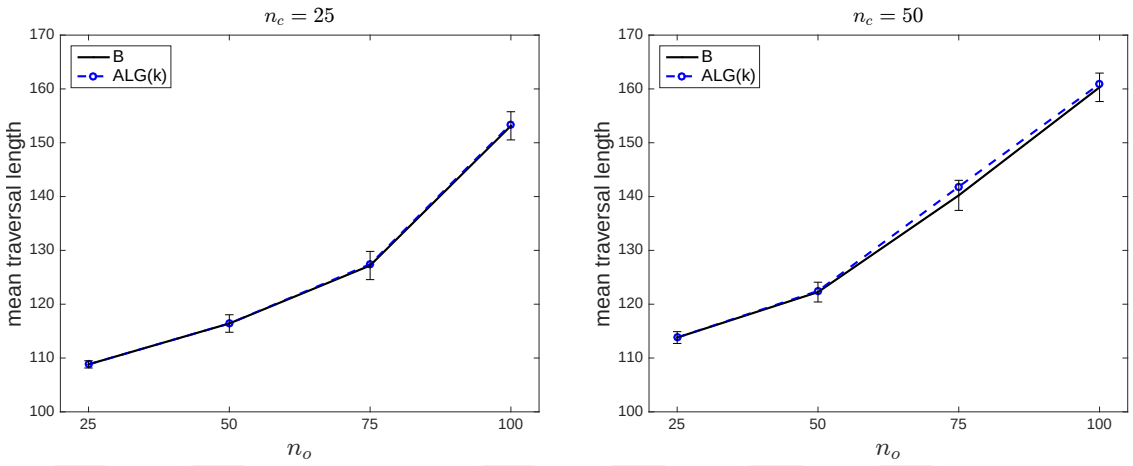


Figure 5.18: Comparison of mean traversal lengths estimated by the B and ALG(k) algorithms, (a) for $n_c = 25$ with $n_o = 25, 50, 75, 100$, (b) and for $n_c = 50$ with $n_o = 25, 50, 75, 100$ with 95% confidence intervals for the benchmark value.

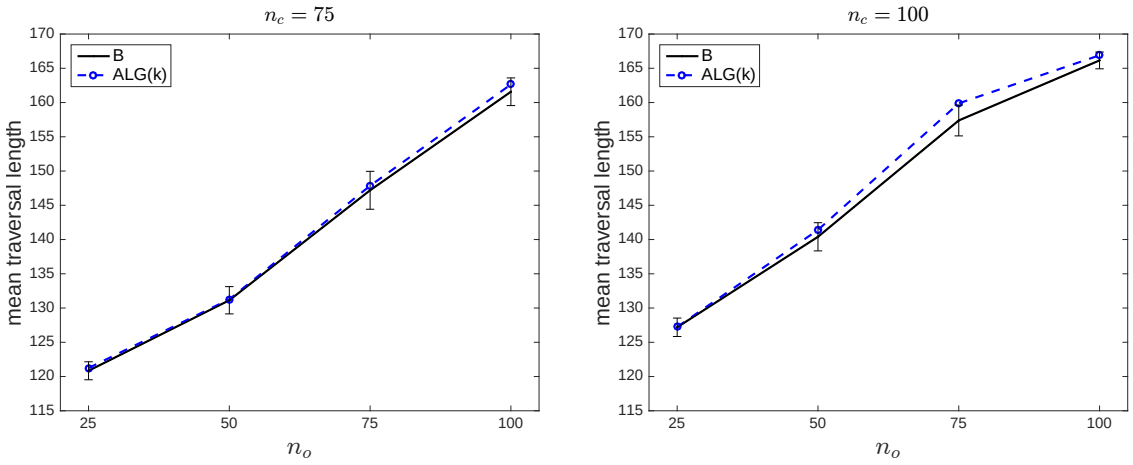


Figure 5.19: Comparison of mean traversal lengths estimated by the B and ALG(k) algorithms, (a) for $n_c = 75$ with $n_o = 25, 50, 75, 100$, (b) and for $n_c = 100$ with $n_o = 25, 50, 75, 100$ with 95% confidence intervals for the benchmark value.

function $\mathcal{F}_k(c, p)$ which approximates the corresponding benchmark value. In some cases, we found almost the same value as the benchmark value and in the worst case (see Figure 5.19(b)), on the average the heuristic traversal length results in less than 1.5% percentage error.

5.8 MT($\tilde{k}$) Algorithm

Algorithm 1: The psuedo code of the penalty function $\mathcal{F}_{MT(\tilde{k})}$.

Input: $X = X_C \cup X_O$

p_i: the non-traversability probability of the disk shaped obstacle D_i

c: the disambiguation of an obstacle

$\tilde{k}$: the range $(0, 2\tilde{k})$ to choose k values

n: the number of subintervals in $(0, 1)$

Output: The penalty function $\mathcal{F}_{MT(\tilde{k})}(c, p)$

Define $n+1$ dimensional arrays $I := \left(0, \frac{1}{n}, \ldots, \frac{n-1}{n}, 1\right)$ and $K := \left(0, \frac{2\tilde{k}}{n}, \ldots, \frac{(n-1)2\tilde{k}}{n}, 2\tilde{k}\right)$

Compute $m_0 = \frac{1}{|X|} \sum_{i=1}^{|X|} p_i$

for $i := 1$ **to** $\frac{n}{2}$ **do**

  if $I(i) \leq m_0 < I(i+1)$ **then**

    $K(i) \leftarrow K(i) + \left(i - \frac{n}{2} - 1\right) \frac{2\tilde{k}}{n}$

  end if

end for

for $i := \frac{n}{2} + 1$ **to** n **do**

  if $I(i) \leq m_0 < I(i+1)$ **then**

    $K(i) \leftarrow K(i) + \left(i - \frac{n}{2}\right) \frac{2\tilde{k}}{n}$

  end if

end for

for $i := 1$ **to** n **do**

  if $K(i) < 0$ **then**

    $K(i) \leftarrow 0$

  else if $K(i) > 2\tilde{k}$ **then**

    $K(i) \leftarrow 2\tilde{k}$

  end if

end for

for $j := 1$ **to** $|X|$ **do**

  for $i := 1$ **to** n **do**

    if $I(i) \leq p_j < I(i+1)$ **then**

      $k \leftarrow K(i)$

      $\mathcal{F}_{MT(\tilde{k})}(c, p_j) \leftarrow \mathcal{F}_k(c, p_j)$

    end if

5.9 Summary

Exploiting penalty functions in current greedy algorithms for NAVA, we managed to unify them in a single family of penalty functions. For each fixed disambiguation cost value c, the penalty function $\mathcal{F}_{RD}(c,p)$ of the RD algorithm, the penalty function $\mathcal{F}_{P}(c,p)$, and the penalty function of the $\mathcal{F}_{DT}(c,p)$ can be approximated by the penalty function of the form $\mathcal{F}_{k}(c,p)$ with $k=2,3,5$, respectively (Corollary 5.3.1.1). And, if ALG(k) is the algorithm to compute the heuristic traversal length demonstrated on the penalty function $\mathcal{F}_{k}(c,p)$, then on the average mean traversal lengths computed by the RD, P, and DT algorithms can be approximated by the mean traversal lengths computed by the algorithms ALG(k) for $k=2,3,5$, consecutively (Proposition 5.4.1). Note that, as k gets larger ($k \geq 5$) the penalty function $\mathcal{F}_{k}(c,p)$ yields larger edge weights, and thus the algorithm ALG(k) becomes reluctant to take risks and avoids disambiguations along the NAVA's traversal. This explains why the DT algorithm avoids disambiguations (unlike the RD and P algorithms) when encountered by the true obstacles on its way, taking no risk, and inclined to choose plain zero-risk path (Figure 5.6).

Let $L_{ALG(k)}(\lambda)$ be the traversal length computed by the ALG(k) algorithm under the sensor precision parameter λ for the given problem setting (see Section 5.4), then in special circumstances such as for the false obstacle only or true obstacle only cases the benchmark value (computed by the algorithm B) can be evaluated exactly by choosing the values $k=0$ and $k \geq 5$, respectively. Moreover, in some special cases we have the stochastic ordering for the traversal length $L_{ALG(k)}(\lambda)$ with respect to both k and sensor precision variable λ (Proposition 5.5.1). For instance, in the false obstacle only case $L_{ALG(k)}(\lambda) \leq_{st} L_{ALG(k)}(\lambda')$ for $\lambda > \lambda'$ and $L_{ALG(k)}(\lambda) \geq_{st} L_{ALG(k')}(\lambda)$ for $k > k'$. And, as the sensor precision tends to perfect detection the traversal length $L_{ALG(k)}(\lambda)$ approaches to the benchmark value L_B (Corollaries 5.5.1.1-5.5.1.2).

The disambiguation cost value c is another important variable in determining the trend of the mean traversal length computed by the corresponding heuristics algorithms. We observed that as the disambiguation cost increases, the mean traversal

length increases up until the algorithm chooses the zero-risk path and performs no disambiguations. The change in disambiguation cost c does not alter the trend in traversal length under fixed sensor precision and same number of obstacles. As the disambiguation cost increases the number of disambiguations carried out by NAVA decreases, hence the traversal length tends to be larger (trend is concave down and logarithmic increase).

Since, all heuristics algorithms are greedy we introduce the idea of disambiguation of an obstacle based on the threshold parameter α, i.e., if the corresponding probability (of being true obstacle) p of any disk shaped obstacle is less than α, then perform the disambiguation, and otherwise bypass the obstacle without taking no risk. Based on this idea, we characterized the penalty function $\mathcal{F}_{T(\alpha)}(c, p)$ and empirically showed that the mean traversal length computed by the algorithm T(0.5) (demonstrated on the $\mathcal{F}_{T(0.5)}$ penalty function) is less than or equal to the mean traversal length computed by the ALG(5) algorithm. On the average the mean traversal length computed by the algorithm T(0.5) does not converge to the benchmark value and the main drawback of the penalty function $\mathcal{F}_{T(0.5)}$ is that it strictly separates clutter and true obstacles from each other by the given threshold $\alpha = 0.5$. In the given problem setting (see Section 5.4), loosening (i.e., decreasing α) or strengthening (i.e., increasing α) the threshold value might cause the loss of clutter or true obstacle information which is not a desirable situation, as it might cause instability in the performance of the algorithm.

To overcome this handicap, we make use of the mean of all probabilities p_i of obstacles, and define the new penalty function $\mathcal{F}_{MT(\tilde{k})}(c, p)$ (see Algorithm 1) and the corresponding MT($\tilde{k}$) algorithm. Simply, it smartly decides which threshold parameter α should be imposed by taking the mean of probabilities into consideration before the traversal of NAVA. Indeed, the MT($\tilde{k}$) algorithm is the improved version of the ALG(k) algorithm by considering the k value as a vector component and taking into account the mean of probabilities of obstacles. Here, if m_0 is the average of probabilities then we divide the unit interval $[0, 1]$ and the interval $[0, 2\tilde{k}]$ into 100 equal parts

(default). Then, we generate the vector K, with dimension equal to the number of obstacles, in accordance with m_0 lying in the same divided subinterval of $[0, 1]$ (see Algorithm 1). For smaller m_0 (such as clutter only case) the components of K tend to be smaller, and for larger m_0 (such as true obstacle only case) the components of K tend to be larger. We showed that in many cases the benchmark value can be approximated within a 1.5% percentage error by using the MT($\tilde{k}$) algorithm for some appropriate choices of $\tilde{k}$ (see Figures 5.16-5.17). On the other hand, the benchmark value can be also approximated by using the ALG(k) algorithm for some positive integer values of k, and this approximated value lies within a 95% confidence interval for the benchmark value (see Figures 5.18-5.19).

In our future research, we will improve our algorithms and show that the benchmark value can be approximated more accurately. The heuristic algorithms that we argued are defined and studied for our problem setting. The penalty functions and the corresponding algorithms can be also applied or modified to solve other problems such as the Canadian traveler's problem of Papadimitriou and Yannakakis (1991), the discrete SOSP of Aksakalli (2007b), the discretized OPD of Aksakalli and Ceyhan (2012), the repeated-task CTP of Bnaya et al. (2015), and the CTP-Tree of Fried (2013). Another aspect field of study of algorithms is the competitive analysis of heuristic algorithms where the ratio of heuristic and worst case performances are investigated (Borodin and El-Yaniv, 1998). The work of Westphal (2008) and Xu et al. (2009) are some competitive analyzes related to our problem. We will analyze the competitive analysis of existing algorithms such as the RD, P and DT algorithms, and carry out similar study for our heuristic ALG(k) and MT($\tilde{k}$) algorithms.

Chapter 6

THEORETICAL RESULTS

6.1 Introduction

As we have discussed in Chapter 5, in the original CTP, the status of an edge (i.e., whether it is blocked or not) can only be revealed upon reaching a vertex incident to that edge. This kind of sensing is called the *local* sensing, and the *remote* sensing refers to revealing the status of an edge from a distant location (Bnaya et al., 2009). The sensing cost in the CTP is equivalent to the disambiguation cost in the D-SOSP.

In the classical CTP problem (Papadimitriou and Yannakakis, 1991), a navigating agent is given a connected graph $G = (V, E)$ together with its initial source $s \in V$, and a target vertex $t \in V$. In addition, each edge $e \in E$ has an Euclidean length $\ell(e)$, and is blocked with a probability $p(e)$, where $p(e)$ is known to the NAVA. Each edge e is either blocked (and hence, not traversable), or unblocked (and hence, traversable at a cost of $\ell(e)$). The status of an edge is remained fix throughout the traversal and is initially unknown to the NAVA. NAVA can learn the status of an edge e only upon reaching the vertex incident on e. Since the exact traversal length is uncertain until the end, the task is to devise a traveling strategy which gives an optimal (or suboptimal) expected traversal length. Considering the similarity between the discrete version of the SOS problem and CTP with sensing, we add a property for NAVA where it can disambiguate an edge e with cost $c(e)$, and learn whether the edge is traversable or not upon arrival at a node incident on e. Thus, an instance of the discrete SOS problem is a 7-tuple $G = (V, E, p, \ell, c, s, t)$, where

- (V, E) is a connected (undirected) graph with vertex set V and edge set E,
- $p : E \longrightarrow [0, 1]$ defines the blocking (non-traversability) probabilities of edges,

- $\ell : E \longrightarrow \mathbb{R}^+$ defines the Euclidean lengths of edges,
- $c : E \longrightarrow \mathbb{R}^+ \cup \{0\}$ defines the disambiguation costs of edges, and
- $s, t \in V$ are the source and target locations, respectively.

For example, if $E = \{e_1, e_2, ..., e_n\}$, then

- $p = \{p_1, p_2, ..., p_n\}$ is the corresponding edge probabilities,
- $\ell = \{\ell_1, \ell_2, ..., \ell_n\}$ is the Euclidean lengths of edges, and
- $c = \{c_1, c_2, ..., c_n\}$ is the disambiguation costs of edges, respectively.

For any discrete SOSP instance $G = (V, E, p, \ell, c, s, t)$ with $|E| = n \in \mathbb{N}$, let the values ℓ, c, s, t are fixed but p changes according to some distribution. Then, it is reasonable to define the expected traversal length from the source s to the target t under the policy π relative to the distribution of p i.e., denoted by $T_n^{G,\pi}(p) = T_n^{\pi}(p)$. That is, $T_n^{\pi}(p)$ is a random variable defined from $[0, 1]^n$ to $\mathbb{R}$ depending on p. Then, formally we are interested in $\min_\pi T_n^{\pi}(p)$ for any fixed n and realization of p. But, in this study we go beyond the scope and investigate the properties of the random variable $T_n^{\pi}(p)$ for some specific distributions of p.

6.2 RD Algorithm for the Discrete SOSP

The reset disambiguation (RD) algorithm introduced in Aksakalli et al. (2011) is optimal for some special types of graphs. Briefly, all edges are assigned by the edge weight function $h(e) = \ell(e) + c(e)/(1 - p(e))$, and then the shortest s, t walk is found using Dijkstra's algorithm. Whenever a probabilistic edge e with probability $p(e) > 0$ is encountered through traversal, NAVA has an option to disambiguate it at a cost $c(e)$ added to the total traversal length, and thus learn whether e is traversable or not. If e is found to be traversable, then NAVA continues its walk through e, otherwise e is removed from the graph. We repeat this procedure until the target point is

reached. For a given discrete SOSP instance $G = (V, E, p, \ell, c, s, t)$, by running the RD algorithm we find the deterministic traversal length denoted by $L(G, RD)$. To perform this action, we add one more information for each edge $e \in E$. Let $b : E \to \{0, 1\}$ be a random variable such that $b(e) = 0$ if e is traversable, and $b(e) = 1$ otherwise. For each instance $G(b) = (V, E, p, b, \ell, c, s, t)$ the random variable b is determined before the traversal depending on p with some decision process or values of b can also be set randomly independent of p. Analogous to $T_n^\pi(p)$, let $L_n^{G(b),\pi}(p) = L_n^\pi(p)$ be the deterministic traversal length from the source s to the target t under the policy π relative to the distribution of p. That is, $L_n^\pi(p)$ is a random variable defined from $[0, 1]^n$ to $\mathbb{R}$ depending on p. Then, formally we are interested in $\min_\pi L_n^\pi(p)$ for any fixed n and realization of p. But, in this study we go beyond the scope and investigate the properties of the random variable $L_n^\pi(p)$ for some specific distributions of p.

6.3 Graphs with Two Nodes

Theorem 6.3.1 *:(Aksakalli (2007b)) Let $G = (V, E, p, \ell, c, s, t)$ be any discrete SOSP instance with $V = \{s, t\}$ and $|E| = n$, then the policy induced by the RD algorithm gives the optimal expected traversal length for any fixed p. Namely,*

$$T_n^{OPT}(p) = \min_\pi T_n^\pi(p) = T_n^{RD}(p). \tag{6.1}$$

In the above theorem, w.l.o.g we can assume that $p_1 = 0, c_1 = 0$, and $p_2, p_3, ..., p_n$ any numbers between 0 and 1. Otherwise, the value of $T_n^\pi(p)$ would be infinite for any policy π.

A *weather* for a CTP instance with edges E is a subset $W \subseteq E$ representing the edges that are traversable in that weather. Weather W is called *good* if s and t remain connected only when using edges in W. Otherwise, W is called *bad* (Eyerich et al. (2010)). Now, let $G = (V, E, p, \ell, c, s, t)$ be any discrete SOSP instance with $V = \{s, t\}$, $|E| = n$, and the values of p are independent and identically distributed (i.i.d) from uniform $(0, 1)$ distribution. We form a *weather* as following; if $p(e) < 0.5$ then set e as traversable, and blocked otherwise. That is, we are forming $G(b)$ instance

from given G instance, i.e., $b(e) = 0$ if $p(e) < 0.5$, and $b(e) = 1$ if $p(e) \geq 0$. By using RD algorithm, we compute the deterministic length $L_n^{RD}(p)$ for this instance. As the number of edges tends to infinity, we claim that the expected traversal length $T_n^{RD}(p)$ and deterministic traversal length $L_n^{RD}(p)$ converge in probability to the same value. To prove our claim, we start with some special cases.

Theorem 6.3.2 *: Let $G = (V, E, p, \ell, c, s, t)$ be any discrete SOSP instance with $V = \{s, t\}$ and $|E| = n$. And suppose that $G(b)$ is a* good weather *induced by the instance G for any realization of p. If*

- *the distribution of p_i for all $i = 2, ..., n$ is i.i.d from uniform $(0, 1)$ distribution with $p_1 = 0$,*
- *the Euclidean distances $\ell_i = 1$, for all $i = 2, ..., n$ with fixed $\ell_1 = a \in \mathbb{R}^+$,*
- *the disambiguation costs $c_i = 5$, for all $i = 2, ..., n$ with $c_1 = 0$, then*

$$T_n^{RD}(p) - L_n^{RD}(p) \xrightarrow{P} 0 \text{ as } n \to \infty. \tag{6.2}$$

Lemma 6.3.2.1 *: Let $G = (V, E, p, \ell, c, s, t)$ be discrete SOSP instance with $V = \{s, t\}$ and $|E| = n$. And suppose that $G(b)$ is a* good weather *induced by the instance G for any realization of p. Under the conditions of Theorem 2, we have*

$$L_n^{RD}(p) = a \text{ for } a \leq 6, \text{ and}$$
$$L_n^{RD}(p) \xrightarrow{P} 6 \text{ as } n \to \infty \text{ for all } a > 6.$$

Proof : By using the weight function $h(e) = \ell(e) + c(e)/(1 - p(e))$ for any edge $e \in E$, we compute $h_1, h_2, ..., h_n$ and sort them in an increasing order, since Dijkstra's algorithm would start with the smallest edge weight. Let π be a permutation of the set $\{1, 2, ..., n\}$ such that

$$h_{\pi(1)} < h_{\pi(2)} < ... < h_{\pi(n)} \tag{6.3}$$

Because of the property of the function h and the given conditions, it is not hard to see that the above ordering always implies the ordering of p_i's for $i = 2, 3, ..., n$, i.e, the order statistics of $p_2, p_3, ..., p_n$. According to the RD policy, the edge order that we are going to follow will be $e_{\pi(1)}, e_{\pi(2)}, ..., e_{\pi(n)}$. For each CTP instance G, we generate *good weather* G(b) instance so as to find the deterministic length. As we have discussed before, each edge is assigned to be traversable if the corresponding edge probability is less than 1/2 and blocked otherwise. Thus, the RD algorithm will terminate when it for the first time encounters $p_{\pi(i)} < 1/2$ for some i. If $a \leq 6$, then $h_1 = a$ and $6 = 1 + 5/(1-0) < \ell_i + c_i/(1-p_i) = h_i$ for all $i = 2, 3, ..., n$. In this case, RD dictates to start with edge e_1 and clearly $L_n^{RD}(p) = a$ for any instance $G(b)$ and realization of p. Now, suppose that $a > 6$. Then, $h_1 = a = h_{\pi(i)}$ for some $i \geq 1$.
Case 1: If $a \leq h_{\pi(1)}$, then RD dictates to start with edge e_1 and clearly $L_n^{RD}(p) = a$. But, this occurs with probability

$$\mathbf{P}\left(a \leq h_{\pi(1)}\right) = \mathbf{P}\left(1 - 5/(a-1) \leq p_{\pi(1)}\right) \tag{6.4}$$

Case 2: If $h_{\pi(1)} < a$ and $p_{\pi(1)} \geq 1/2$, then RD dictates to start with edge $e_{\pi(1)}$ and continues until it reaches the edge $e_{\pi(i)} = e_1$ because in this case all $p_i \geq 1/2$ for all $i = 2, 3, ..., n$. So, $L_n^{RD}(p)$ equals to $\ell_{\pi(i)} + \sum_{k=1}^{i} c_{\pi(k)} = a + (i) \times 5 = 5i + a$ where $i \geq 2$. But, this occurs with probability

$$\mathbf{P}\left(h_{\pi(1)} < a \text{ and } p_{\pi(1)} \geq 1/2\right) = \mathbf{P}\left(1/2 \leq p_{\pi(1)} < 1 - 5/(a-1)\right) \tag{6.5}$$

Case 3: If $h_{\pi(1)} < a$ and $p_{\pi(1)} < 1/2$, then RD dictates to start with edge $e_{\pi(1)}$ and clearly $L_n^{RD}(p) = \ell_{\pi(1)} + c_{\pi(1)} = 6$. And, this occurs with probability

$$\mathbf{P}\left(h_{\pi(1)} < a \text{ and } p_{\pi(1)} < 1/2\right) = \mathbf{P}\left(p_{\pi(1)} < \min\left(1 - 5/(a-1), 1/2\right)\right). \tag{6.6}$$

It suffices to show that the probabilities in (6.4) and (6.5) converges to zero as n tends to infinite. Observe that $p_{\pi(1)}$ is the minimum of $p_2, p_3, ..., p_n$, so we know explicitly its probability density function which depends on n. It is well known that the k^{th} order statistics of n i.i.d uniform $(0, 1)$ distributed random variables is Beta$(k, n + 1 - k)$

distribution. So, $p_{\pi(1)}$ has probability density function $(1-x)^{n-2} \times (n-1)$. The probability in (6.4) becomes,

$$\begin{aligned}
\mathbf{P}\left(a \leq h_{\pi(1)}\right) &= \mathbf{P}\left(1 - 5/(a-1) \leq p_{\pi(1)}\right) \\
&= \int_{1-5/(a-1)}^{1} (n-1)(1-x)^{n-2} dx \\
&= \left(\frac{5}{a-1}\right)^{n-1} \longrightarrow 0 \text{ as } n \to \infty.
\end{aligned}$$

And the probability in (6.5) becomes,

$$\begin{aligned}
\mathbf{P}\left(h_{\pi(1)} < a \text{ and } p_{\pi(1)} \geq 1/2\right) &= \mathbf{P}\left(1/2 \leq p_{\pi(1)} < 1 - 5/(a-1)\right) \\
&= \int_{1/2}^{1-5/(a-1)} (n-1)(1-x)^{n-2} dx \\
&= \left(\frac{1}{2}\right)^{n-1} - \left(\frac{5}{a-1}\right)^{n-1} \longrightarrow 0 \text{ as } n \to \infty.
\end{aligned}$$

which completes the proof of Lemma 6.3.2.1.

On the other hand, in order to compute the expected traversal length induced by RD algorithm we no longer need *good weather*. Since, we are not interested in the deterministic value of traversal length.

Lemma 6.3.2.2 *: Let $G = (V, E, p, \ell, c, s, t)$ be any discrete SOSP instance with $V = \{s, t\}$ and $|E| = n$. Under the conditions of Theorem 2, we have*

$$T_n^{RD}(p) = a \text{ for } a \leq 6, \text{ and} \tag{6.7}$$

$$T_n^{RD}(p) \xrightarrow{P} 6 \text{ as } n \to \infty \text{ for all } a > 6. \tag{6.8}$$

Proof : We are going to use the same ordering induced by *RD* algorithm given in (6.3). Thus, *RD* dictates to start with edge $e_{\pi(1)}$ and so on. In this case, the computation will be terminated whenever e_1 is encountered. For the discrete SOSP instance, actually we have an explicit form of the expected traversal length given in Aksakalli (2007b). Namely,

$$T_n^{RD}(p) = \sum_{i=1}^{n} \left(\prod_{j<i} p_{\pi(j)}\right) \left(1 - p_{\pi(i)}\right) h_{\pi(i)} \text{ where } p_{\pi(0)} = 1. \tag{6.9}$$

Moreover, this expression gives the optimal expected traversal length for any realization of p (Theorem 6.3.1). If $a \leq 6$, then $a = h_1 \leq 6 < h_i$ for $i = 2, 3, ..., n$. So, in the ordering (6.3) we get $\pi(1) = 1$. Since, $p_1 = p_{\pi(1)} = 0$, $\prod_{j<i} p_{\pi(j)} = 0$ for all $i \geq 2$. In this case, $T_n^{RD}(p)$ becomes

$$\begin{aligned} T_n^{RD}(p) &= \sum_{i=1}^{1} \left(\prod_{j<i} p_{\pi(j)} \right) \left(1 - p_{\pi(i)}\right) h_{\pi(i)} \\ &= p_{\pi(0)} \left(1 - p_{\pi(1)}\right) h_{\pi(1)} \\ &= a. \end{aligned}$$

Which proves the first part of Lemma 2. Now, suppose that $a > 6$.

Case 1: If $a \leq h_{\pi(1)}$, then RD dictates to start with edge e_1 and clearly $T_n^{RD}(p) = a$. But, this occurs with probability

$$\mathbf{P}\left(a \leq h_{\pi(1)}\right) = \mathbf{P}\left(1 - 5/(a-1) \leq p_{\pi(1)}\right) \tag{6.10}$$

which converges to zero as n tends to infinity (6.4).

Case 2: If $h_{\pi(1)} < a$, then there exist $k = k(n) \geq 2$ such that $p_{\pi(k)} = 0$. That is to say, the value h_1 occurs between $(k-1)^{th}$ and $(k+1)^{th}$ indices in the ordering (6.3).

$$h_{\pi(1)} < h_{\pi(2)} < ... < h_{\pi(k-1)} < h_1 = h_{\pi(k)} < h_{\pi(k+1)} < ... < h_{\pi(n)} \tag{6.11}$$

Since the ordering of h_i does not change the ordering of p_i, the above inequality implies that

$$p_{\pi(1)} < p_{\pi(2)} < ... < p_{\pi(k-1)} < 1 - 5/(a-1) < p_{\pi(k+1)} < ...p_{\pi(n)} \tag{6.12}$$

Observe that the index k may change as n changes for fixed value of a. In this case RD dictates to start with edge $e_{\pi(1)}$ and continues until it reaches the edge $e_{\pi(k)} = e_1$ because $\prod_{j<i} p_{\pi(j)} = 0$ for $i > k$. So, $T_n^{RD}(p)$ becomes,

$$\begin{aligned} T_n^{RD}(p) &= \sum_{i=1}^{k} \left(\prod_{j<i} p_{\pi(j)} \right) \left(1 - p_{\pi(i)}\right) h_{\pi(i)} \\ &= \sum_{i=1}^{k-1} \left(\prod_{j<i} p_{\pi(j)} \right) \left(6 - p_{\pi(i)}\right) + \left(\prod_{j<k} p_{\pi(j)} \right) (1 - p_{\pi(k)}) h_{\pi(k)} \\ &= 6 + 5 \sum_{i=1}^{k-2} \prod_{j=1}^{i} p_{\pi(j)} + (a-1) \prod_{j=1}^{k-1} p_{\pi(j)}. \end{aligned}$$

From (6.12) and $5 < a - 1$, we have the following inequality,

$$\begin{aligned} T_n^{RD}(p) &\leq 6 + (a-1) \sum_{i=1}^{k-1} \prod_{j=1}^{i} p_{\pi(j)} \\ &\leq 6 + (a-1) p_{\pi(1)} \sum_{i=0}^{k-2} \left(1 - \frac{5}{a-1}\right)^i \\ &\leq 6 + p_{\pi(1)} (a-1)^2 / 5 \end{aligned}$$

This upper bound is independent of k. Since, $p_{\pi(1)}$ converges in probability to zero as n tends to infinity, we deduce that $T_n^{RD}(p)$ converges in probability to 6 as n tends to infinity, which completes the proof of Lemma 2.

Corollary 6.3.0.1 *(of Lemma 6.3.2.2): For any fixed a, we have that $T_n^{RD}(p) < C(a)$ where the constant $C(a)$ only depends on a. Indeed, $C(a) = a$ for $a \leq 6$, and $C(a) = 6 + (a-1)^2/5$ for $a > 6$. For $a < 6$, from Lemma 6.3.2.2 and the Dominated Convergence Theorem it follows that $\mathbf{E}T_n^{RD}(p)$ converges to 6 as n tends to infinity.*

Proof of Theorem 6.3.2: By combining Lemma 6.3.2.1 and Lemma 6.3.2.2, and using the Slutsky's theorem the result follows.

As we have discussed in Chapter 5, both ALG(k) and MT($\tilde{k}$) might be used as a heuristic algorithms in discrete SOS problem. So, we have the following conjecture;

Conjecture: *Let $G = (V, E)$ be a graph with $|E| = m$ and $s, t \in V$ be given source and target points, respectively. Let $E' \subset E$ be the probabilistic edges (in a grid these are edges intersecting the obstacles) with $|E'| = m'$. For each edge $e \in E$, denote by $l(e), c(e), p(e)$ the Euclidean distance of edge e, the disambiguation cost of edge e, and the probability of non-traversability of edge e, respectively. Thus, $c(e) = 0$ and $p(e) = 0$ for $e \in E - E'$. For any given problem instance with fixed sensor precision parameter λ, let L_B is the traversal length of benchmark value, $L_{ALG(k)}(\lambda)$ is the traversal length computed by the ALG(k) algorithm, and $L_{MT}(\tilde{k}, \lambda)$ be the traversal length computed by the MT($\tilde{k}$) algorithm, then*

$$\mathbf{E}L_{ALG(k)}(\lambda) \approx \mathbf{E}L_B \text{ for some } k,$$

and,

$$\mathbf{E}L_{MT}(\tilde{k}, \lambda) \approx \mathbf{E}L_B \text{ for some } \tilde{k}.$$

6.3.1 Empirical Results

Under the conditions of Theorem 6.3.2, we have carried out 100 Monte Carlo simulation for each value of a and n. The values of a are $6, 8, 10, 20, 30$, and the values of n are $10, 100, 1000, 10000$. For each combined treatment of a and n, we have generated the values of p from uniform $(0, 1)$ distribution 100 times, and exhibit the average values of corresponding $L_n^{RD}(p)$ and $T_n^{RD}(p)$ in Table 6.1.

6.4 Generalization of OPD Problem

Another aspect of OPD problem is considering the study window in a higher dimensional setting. That is to say, we would like to investigate the OPD problem in higher

		Mean of $T_n^{RD}(p)$	Mean of $L_n^{RD}(p)$
$a = 6$	$n = 10$	6	6
	$n = 100$	6	6
	$n = 1000$	6	6
	$n = 10000$	6	6
$a = 8$	$n = 10$	6.625949	6.08
	$n = 100$	6.041687	6.00
	$n = 1000$	6.005609	6.00
	$n = 10000$	6.000471	6.00
$a = 10$	$n = 10$	6.740787	6.00
	$n = 100$	6.049613	6.00
	$n = 1000$	6.005848	6.00
	$n = 10000$	6.000658	6.00
$a = 20$	$n = 10$	6.889451	6.00
	$n = 100$	6.049774	6.00
	$n = 1000$	6.005096	6.00
	$n = 10000$	6.000451	6.00
$a = 30$	$n = 10$	6.880909	6.54
	$n = 100$	6.044417	6.00
	$n = 1000$	6.005060	6.00
	$n = 10000$	6.000458	6.00

Table 6.1: For any fixed values of a and n, there are 100 realizations of values of p selected randomly from uniform $(0, 1)$ distribution. And, we exhibit the corresponding mean values.

dimension as well starting from three dimensional case. In 3-D case, NAVA may represent the submarine that wishes to reach from one source point to a target destination avoiding underwater mines. And, OPA may be the opponent forces wishing

to place mines so as to maximize the total traversal length of NAVA.

6.4.1 Higher Dimensional Discretization

We want to discretize any finite subset G of $\mathbb{R}^d$ where $d \geq 1$. Let G be a graph with vertices are integer lattices and edges be usual lattice edges. Additionally, in any dimension d, draw edges between all pairs of vertices of G such that the Euclidean norm between these pair of vertices is at most $\sqrt{d}$ units. Without loss of generality, suppose that G contains the origin of R^d, and under this construction, let k be the number of edges in G connected to the origin. This is called the k-adjacency integer lattice discretization, and it is the generalizaiton of the 8-adjacency integer lattice discretization given in Aksakalli et al. (2011). For any interior point x of $G \cap \mathbb{Z}^d$, there are exactly k lattice points incident to v, that is where the name is coming from.

Example 6.4.1 *: Let $d = 1$, then Z^1 is an axis line and clearly there are only 2 edges in G connected to the origin, and the Euclidean norm from the origin to these vertices is at most $\sqrt{1}$ units. Namely, the vertices {1,-1}. Respective Euclidean norms are $\|0-1\|^2 = \sqrt{1}$ and $\|0-(-1)\|^2 = \sqrt{1}$.*

Example 6.4.2 *: If $d = 2$, then Z^2 is an usual square lattice and clearly there are only 8 edges in G connected to the origin and the Euclidean norm from the origin to these vertices is at most $\sqrt{2}$ units. Namely, the vertices {(1,0),(0,1),(-1,0),(0,-1)} with 1 units apart, and the vertices {(1,1),(-1,1),(-1,-1),(1,-1)} with $\sqrt{2}$ units apart from the origin. Sample computations are $\|(0,0)-(1,0)\|^2 = 1$ and $\|(0,0)-(1,1)\|^2 = 2$. This is the 8-adjacency integer lattice discretization (Aksakalli et al. (2011)).*

In d dimensional case, indeed k is the number of integer lattice points inside a sphere of radius $\sqrt{d}$ centered at the origin. For small values of d, there is a closed formula and we are going to present some corresponding values of k. As for large values of d,

there is a nice asymptotic result.

Proposition 6.4.1 *(Nguyen and Stern (2005)) : Let $N(n,r)$ be the number of integer points in the n-dimensional sphere of radius $\sqrt{r}$ centered at the origin: that is, $N(n,r)$ is the number of $(x_1, x_2, ..., x_n) \in \mathbb{Z}^n$ such that $\sum_{i=1}^{d} x_i^2 \leq r$. Then, we have the following induction formula:*

$$N(n,r) = \sum_{j=-\lfloor\sqrt{r}\rfloor}^{\lfloor\sqrt{r}\rfloor} N(n-1, r-j^2). \tag{6.13}$$

This allows to compute $N(n,r)$ numerically when n and r is are not too large, since the running time is clearly polynomial in (n,r) (see Table 6.2). In our case, we are interested in the value of $N(n,r)$ when $n = d$ and $r = d$.

Corollary 6.4.1.1 *: In d dimension, $k = N(d,d) - 1$ with $N(1,1) = 3$ and $N(2,2) = 9$.*

Proof of **Proposition 6.5.1** (Galante (2005)): When $n = 1$, then just look at the number of integer points in an interval about the origin. Note that from 0 to $\sqrt{r}$ (excluding zero) there are $\lfloor\sqrt{r}\rfloor$ points and similarly from 0 to $-\sqrt{r}$ (excluding zero) there are $\lfloor\sqrt{r}\rfloor$ points , where $\lfloor\sqrt{r}\rfloor$ denotes the usual greatest integer function. So, counting the origin, there are

$$N(1,r) = 2\lfloor\sqrt{r}\rfloor + 1. \tag{6.14}$$

points in the interval.

When $n = 2$, at each integer lattice j on the x-axis inside the circle, there is an interval of radius $\sqrt{r-j^2}$, and from (6.14) we know how to count the numnber of points in this interval. The j values we are counting will range from $\lfloor\sqrt{r}\rfloor$ to $-\lfloor\sqrt{r}\rfloor$. So that,

$$N(2,r) = \sum_{j=-\lfloor\sqrt{r}\rfloor}^{\lfloor\sqrt{r}\rfloor} N(1, r-j^2) = \sum_{j=-\lfloor\sqrt{r}\rfloor}^{\lfloor\sqrt{r}\rfloor} 2\left\lfloor\sqrt{r-j^2}\right\rfloor + 1. \tag{6.15}$$

which will give the exact solution in $\mathbb{Z}^2$.

Now, we can use the similar logic and generalize the the induction formula for higher dimensions. In $\mathbb{Z}^n$, the problem becomes counting the number of points $(x_1, x_2, ..., x_n)$ with all the x_i integers in the n-dimensional sphere of radius $\sqrt{r}$ centered at the origin. But, just as in Equation (6.13), we observe that the points in the n-dimensional sphere are arranged so that we can count them in $(n-1)$ dimension sphere of radius $\sqrt{r-x_n^2}$ with x_n ranging from $-\lfloor\sqrt{r}\rfloor$ to $\lfloor\sqrt{r}\rfloor$. So that,

$$N(n,r) = \sum_{j=-\lfloor\sqrt{r}\rfloor}^{\lfloor\sqrt{r}\rfloor} N(n-1, r-j^2) \text{ with } N(1,r) = 2\left\lfloor\sqrt{r}\right\rfloor + 1. \tag{6.16}$$

which completes the proof of the proposition.

dimension=d	radius=$\sqrt{d}$	$k = N(d,d) - 1$
1	1	2
2	$\sqrt{2}$	8
3	$\sqrt{3}$	26
4	2	88

Table 6.2: Some values of k in d-dimensional case

When n grows to infinity, sharp estimates of $N(n,r)$ are known when r is proportional to n (Mazo and Odlyzko (1990)), in which case $N(n,r)$ is exponential in n.

6.5 Future Research

In our future research, we will try to prove our results for more general graphical settings by inspiring from current theoretical results belonging to CTP problem. In OPD problem only disambiguation capability was considered, but we also like to consider the problem with neutralization capability. Under feasible conditions, we will study the OPD problem by both disambiguation and neutralization capabilities together depending on the circumstances that NAVA and OPA encounters.

We will also improve our algorithms and show that the benchmark value can be approximated more accurately. The heuristic algorithms that we argued are defined and studied for our problem setting. The penalty functions and the corresponding algorithms can be also applied or modified to solve other problems such as the Canadian traveler's problem of Papadimitriou and Yannakakis (1991), the discrete SOSP of Aksakalli (2007b), the discretized OPD of Aksakalli and Ceyhan (2012), the repeated-task CTP of Bnaya et al. (2015), and the CTP-Tree of Fried (2013). Another aspect field of study of algorithms is the competitive analysis of heuristic algorithms where the ratio of heuristic and worst case performances are investigated (Borodin and El-Yaniv, 1998). The work of Westphal (2008) and Xu et al. (2009) are some competitive analyzes related to our problem. We will analyze the competitive analysis of existing algorithms such as the RD, P and DT algorithms, and carry out similar study for our heuristic ALG(k) and MT($\tilde{k}$) algorithms. Concerning the discrete SOSP, we will try to prove our Conjecture stated in the previous section.

Chapter 7

CONCLUSIONS AND DISCUSSION

First, we investigated how the traversal length of NAVA changes as the obstacle pattern changes from uniformness to regularity, and from uniformness to clustering for various pairs of numbers of clutter and true obstacles n_c, n_o. In our experimental setting, we simulated various combinations of both false and true obstacle number levels. Based on our investigation with extensive Monte Carlo simulations, we found that traversal lengths are higher under regularity than those under uniformness which tend to be higher that those under clustering.

Under regularity, we investigate the influence of the two parameters d and γ. Given the number of obstacles, the γ is the intensity of the number of pairs of distinct points lying closer than d units. Given the total number of obstacles (i.e., $n = n_c + n_o$ is fixed), the ratio (ρ) of true versus false obstacles is another important factor for the traversal length. In all cases, we recommend choosing moderate values of d (around 7) together with small values of γ (less than 0.1) and choosing ρ value as large as possible in order to maximize the total traversal length of NAVA. Under clustering obstacle pattern, the cluster radius r_0 and the number of parent points κ are found to be important. Here, the κ stands for the number of accumulation points and r_0 stands for the radius at which obstacles accumulate around the accumulation point. We do not recommend choosing clustering point pattern since it is not feasible.

When the obstacle pattern and the number of obstacles kept same, then the traversal length under cluttering pattern is stochastically less than or equal to the traversal length under uniformness, and which is also stochastically less than or equal to the traversal length under regularity. Thus, mean traversal length is maximized then the obstacle pattern is regular. Moreover, the traversal length in clutter only case

is stochastically less than or equal to the traversal length in mixture case which is also stochastically less than or equal to the traversal length in true obstacle case only. Concerning the precision of sensor, as the sensor of NAVA increases the traversal length tends to be the more accurate and yields closer value to optimal solution.

We also investigated the differences in trends of traversal length of NAVA computed by M16, M8, and M4 algorithms compared to the traversal length computed by RD algorithm carried out by Aksakalli and Ceyhan (2012). Simply, M2^k algorithm is based on the effective choice of number of disambiguation, and the main advantage of using this algorithm is the reduced complexity time with nice approximation of traversal lengths within 2.5% relative error. To gain more power and precision in our analysis, we used similar statistical analysis tools, i.e., the repeated measures of ANOVA, and extensive Monte Carlo simulations indicate that the trends of mean traversal lengths are essentially similar for each M16, M8, M4 algorithms compared to RD algorithm.

In the OPD problem, on the behalf of OPA, we wish the traversal length of NAVA to be as larger as possible. To achieve this, we choose the clutter pattern as regular as possible i.e., the background clutter pattern is Hardcore(n_c, d) where n_c is the number of false obstacles, and d represents the distance such that there is no pairwise disk-shaped obstacles closer than d units (see Chapter 2). Together with this setting, true obstacles are uniformly distributed inside various obstacle forms such as linear strip, V-shaped, and semicircular, and elliptical obstacle window types. Note that, to attain consistent results the total area covered by obstacle forms are fixed to 800 units. Based on our Monte Carlo simulations, we observe that in overall comparison the traversal length is maximized when obstacle type is the elliptical obstacle window locating close to the source point.

Next, we investigated the case when the background clutter type is again of Hardcore type but true obstacles are distributed as regular as possible inside obstacle window types rather than uniform. One of advantages of studying this situation is that the traversal length might be maximized using fewer true obstacles. For in-

stance, when we consider the elliptical obstacle window type as shown in Figure 4.13, the mean traversal length tends to be larger compared to standard SC90 obstacle form. The mean traversal length is about 230 when $n_o = 60$ true obstacles are uniformly distributed (Figure 4.14), and 465 units when $n_o = 30$ true obstacles are regularly distributed (Figure 4.19). Hence, the mean traversal length is maximized when 30 true obstacles are distributed regularly (i.e., Hardcore($n_o = 30, d_2 = 6$)) inside the elliptical obstacle form as shown in Figure 4.13 ($u = 20$) together with Hardcore($n_c = 100, d$) background clutter type for $d = 5, 6, 7, 8$.

On the other hand, we observed that knowing the exact locations of obstacles does not provide significant gain for NAVA more advantage than knowing only the distribution of obstacles. So we do not recommend using tessellations based on the allocation of clutters so as to maximize the traversal length.

Exploiting penalty functions in current greedy algorithms for NAVA, we managed to unify them in a single family of penalty functions. For each fixed disambiguation cost value c, the penalty function $\mathcal{F}_{RD}(c, p)$ of the RD algorithm, the penalty function $\mathcal{F}_P(c, p)$, and the penalty function of the $\mathcal{F}_{DT}(c, p)$ can be approximated by the penalty function of the form $\mathcal{F}_k(c, p)$ with $k = 2, 3, 5$, respectively (Corollary 5.3.1.1). And, if ALG(k) is the algorithm to compute the heuristic traversal length demonstrated on the penalty function $\mathcal{F}_k(c, p)$, then on the average mean traversal lengths computed by the RD, P, and DT algorithms can be approximated by the mean traversal lengths computed by the algorithms ALG(k) for $k = 2, 3, 5$, consecutively (Proposition 5.4.1). Note that, as k gets larger ($k \geq 5$) the penalty function $\mathcal{F}_k(c, p)$ yields larger edge weights, and thus the algorithm ALG(k) becomes reluctant to take risks and avoids disambiguations along the NAVA's traversal. This explains why the DT algorithm avoids disambiguations (unlike the RD and P algorithms) when encountered by the true obstacles on its way, taking no risk, and inclined to choose plain zero-risk path (Figure 5.6).

Since, all heuristics algorithms are greedy we introduce the idea of disambiguation of an obstacle based on the threshold parameter α, i.e., if the corresponding probabil-

ity (of being true obstacle) p of any disk shaped obstacle is less than α, then perform the disambiguation, and otherwise bypass the obstacle without taking no risk. Based on this idea, we characterized the penalty function $\mathcal{F}_{T(\alpha)}(c,p)$ and empirically showed that the mean traversal length computed by the algorithm T(0.5) (demonstrated on the $\mathcal{F}_{T(0.5)}$ penalty function) is less than or equal to the mean traversal length computed by the ALG(5) algorithm. On the average the mean traversal length computed by the algorithm T(0.5) does not converge to the benchmark value and the main drawback of the penalty function $\mathcal{F}_{T(0.5)}$ is that it strictly separates clutter and true obstacles from each other by the given threshold $\alpha = 0.5$. In the given problem setting (see Section 5.4), loosening (i.e., decreasing α) or strengthening (i.e., increasing α) the threshold value might cause the loss of clutter or true obstacle information which is not a desirable situation, as it might cause instability in the performance of the algorithm.

To overcome this handicap, we make use of the mean of all probabilities p_i of obstacles, and define the new penalty function $\mathcal{F}_{MT(\tilde{k})}(c,p)$ (see Algorithm 1) and the corresponding MT($\tilde{k}$) algorithm. Simply, it smartly decides which threshold parameter α should be imposed by taking the mean of probabilities into consideration before the traversal of NAVA. Indeed, the MT($\tilde{k}$) algorithm is the improved version of the ALG(k) algorithm by considering the k value as a vector component and taking into account the mean of probabilities of obstacles. Here, if m_0 is the average of probabilities then we divide the unit interval $[0,1]$ and the interval $[0,2\tilde{k}]$ into 100 equal parts (default). Then, we generate the vector K, with dimension equal to the number of obstacles, in accordance with m_0 lying in the same divided subinterval of $[0,1]$ (see Algorithm 1). For smaller m_0 (such as clutter only case) the components of K tend to be smaller, and for larger m_0 (such as true obstacle only case) the components of K tend to be larger. We showed that in many cases the benchmark value can be approximated within a 1.5% percentage error by using the MT($\tilde{k}$) algorithm for some appropriate choices of $\tilde{k}$ (see Figures 5.16-5.17).

BIBLIOGRAPHY

Aksakalli, V. (2007a). The BAO* algorithm for stochastic shortest path problem with dynamic learning. In *Proceedings of IEEE Conference on Decision and Control*, New Orleans, LA.

Aksakalli, V. (2007b). *Protocols for stochastic shortest path problems with dynamic learning*. PhD thesis, The Johns Hopkins University.

Aksakalli, V. and Ari, I. (2014). Penalty-based algorithms for the stochastic obstacle scene problem. *INFORMS J. Comput.*, 26(2):370–384.

Aksakalli, V. and Ceyhan, E. (2012). Optimal obstacle placement with disambiguations. *The Annals of Applied Statistics*, 6(4):1730–1774.

Aksakalli, V., Fishkind, D. E., Priebe, C. E., and Ye, X. (2011). The reset disambiguation policy for navigating stochastic obstacle fields. *Naval Research Logistics*, 58(4):389–399.

Aksakalli, V., Sahin, O. F., and Ari, I. (2016). An AO* based exact algorithm for the Canadian traveler problem. *INFORMS J. Comput.*, 28(1):96–111.

Baddeley, A. (2010). Analysing spatial point patterns in R. *Workshop notes Ver. 4.1.*

Baddeley, A. et al. (2008). Analysing spatial point patterns with R. In *Workshop Notes*, Australia. CSIRO and University of Australia.

Bar-Noy, A. and Schieber, B. (1991). The Canadian traveller problem. In *Proceedings of the Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 261–270, San Francisco, CA.

Bnaya, Z., Felner, A., Fried, D., Maksin, O., and Shimony, S. E. (2015). Repeated-task Canadian traveler problem. *Aartificial Intelligence Communications*, 28(3):453–477.

Bnaya, Z., Felner, A., Shimony, E., Kaminka, G. A., and Merdler, E. (2008). A fresh look at sensor-based navigation: Navigation with sensing costs. In *AAAI 2008 Workshop on Search in Artificial Intelligence and Robotics*, pages 11–17, Chicago, IL.

Bnaya, Z., Felner, A., and Shimony, S. E. (2009). Canadian traveler problem with remote sensing. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 437–442, Pasadena, CA.

Borodin, A. and El-Yaniv, R. (1998). *Online computation and competitive analysis*. Cambridge University Press, New York.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to algorithms*. MIT Press, Cambridge, MA, third edition.

Eyerich, P., Keller, T., and Helmert, M. (2010). High-quality policies for the Canadian traveler's problem. In *the 24th AAAI Conference on Artificial Intelligence*, Atlanda, GA.

Fishkind, D. E., Priebe, C. E., Giles, K., Smith, L. N., and Aksakalli, V. (2007). Disambiguation protocols based on risk simulation. *IEEE Transactions on Systems, Man, and Cybernetics*, 37(5):814–823.

Fried, D. (2013). *Theoretical Aspects of the Generalized Canadian Traveler Problem*. PhD thesis, Ben-Gurion University of the Negev, Beersheba, Israel.

Fried, D., Shimony, S. E., Benbassat, A., and Wenner, C. (2013). Complexity of Canadian traveler problem variants. *Theoret. Comput. Sci.*, 487:1–16.

Galante, J. (2005). Gauss's circle problem. *Senior Thesis, University of Rochester*.

Mazo, J. E. and Odlyzko, A. M. (1990). Lattice points in high-dimensional spheres. *Monatsh. Math*, 100:47–61.

Nguyen, P. Q. and Stern, J. (2005). Adapting density attacks to low-weight knapsacks. In *Advances in Crptology-ASIACRYPT*, pages 41–58, Chennai, India.

Nikolova, E. and Karger, D. R. (2008). Route planning under uncertainty: The Canadian traveller problem. In *the 23rd AAAI Conference on Artificial Intelligence*, Chicago, IL.

Okabe, A., Boots, B., Sugihara, K., and Chiu, S. N. (2000). *Spatial tessellations: concepts and applications of Voronoi diagrams.* Wiley Series in Probability and Statistics. John Wiley & Sons, Ltd., Chichester, second edition. With a foreword by D. G. Kendall.

Papadimitriou, C. H. and Yannakakis, M. (1991). Shortest paths without a map. *Theoretical Computer Science*, 84(1):127–150.

Priebe, C. E., Fishkind, D. E., Abrams, L., and Piatko, C. D. (2005). Random disambiguation paths for traversing a mapped hazard field. *Naval Research Logistics*, 52(3):285–292.

Seltman, H. J. (2012). Experimental design and analysis. *Online at: http://www. stat. cmu. edu/, hseltman/309/Book/Book. pdf.*

Westphal, S. (2008). A note on the k-Canadian traveller problem. *Inform. Process. Lett.*, 106(3):87–89.

Xu, Y., Hu, M., Su, B., Zhu, B., and Zhu, Z. (2009). The Canadian traveler's problem and its competitive analysis. *J. Comb. Optim.*, 18(2):195–205.

Ye, X., Fishkind, D. E., Abrams, L., and Priebe, C. E. (2011). Sensor information monotonicity in disambiguation protocols. *Journal of the Operational Research Society*, 62(1):142–151.